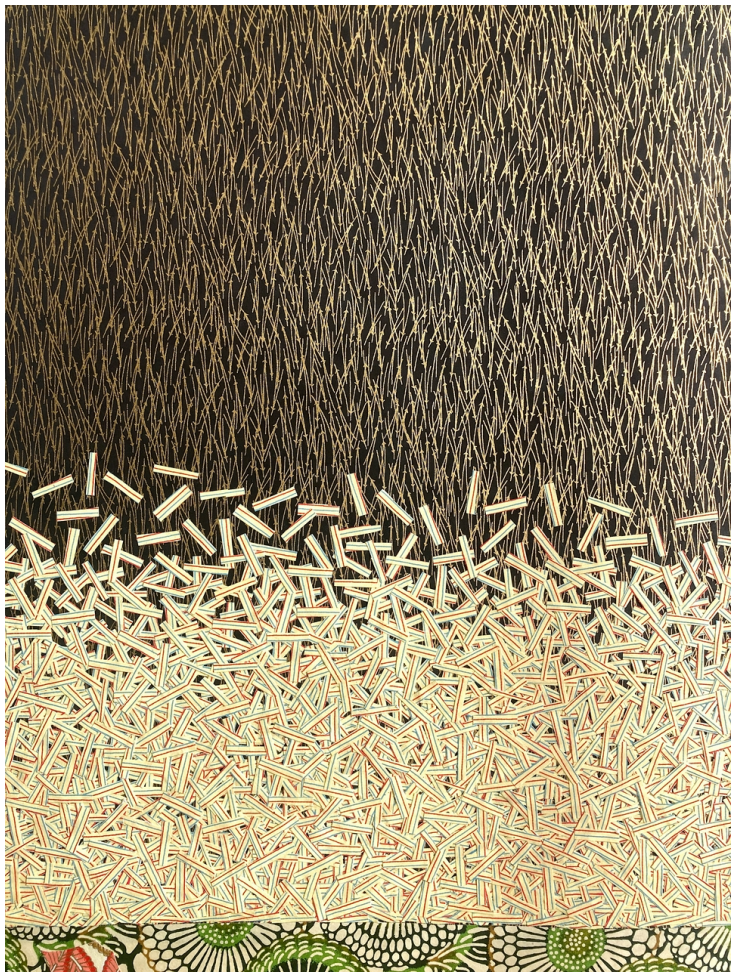
mornings with chair friend

Ocean Beach (2022-23)

Ocean Beach (2022-23)



Chiyogami and paper on wood. 12 x 36 inches. This piece is **availble for sale** https://or-this.myshopify.com/products/ocean-beach-2022-23? .

2023-01-17

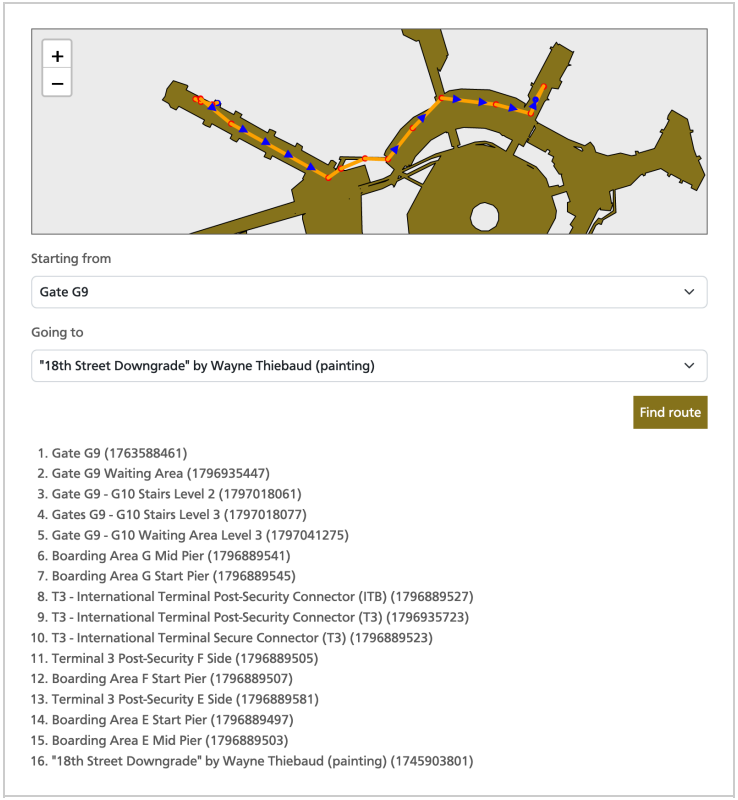
things I have written about elsewhere #20230214

The Airport is the Museum - An Experimental...

The Airport is the Museum - An Experimental Wayfinding Service for SFO Museum

This was **originally published on the SFO Museum Mills Field weblog** https://millsfield.sfomuseum.org/blog/2023/02/14/wayfinding/ , in February 2023.

Today we are releasing an **experimental web-based wayfinding service** https://millsfield.sfomuseum.org/wayfinding/ for SFO Museum. Using this service you can ask for directions between two "waypoints" at SFO. For example, traveling from **Gate G9** https://millsfield.sfomuseum.org/gates/G9 to Wayne Thiebaud's painting **18th Street Downgrade** https://www.sfomuseum.org/public-art/public-collection/18th-street-downgrade in **Boading Area E** https://millsfield.sfomuseum.org/boardingareas/1763588335/ :



https://millsfield.sfomuseum.org/wayfinding/#from=1763588461&to=1745903801

Or from the Thiebaud painting, in **Terminal 3** `https://millsfield.sfomuseum.org/terminals/1763588269/` , to Yayoi Kusama's sculpture **High Heels for Going To Heaven** `https://www.sfomuseum.org/public-art/public-collection/high-heels-going-heaven` in **Terminal 1** `https://millsfield.sfomuseum.org/terminals/1763588175/` :

https://millsfield.sfomuseum.org/wayfinding/#from=1745903801&to=1745903903

Or maybe you are waiting for a passenger at the Boarding Area E **Meet and Greet area** `https://millsfield.sfomuseum.org/galleries/1763595139/` and their flight is delayed and you'd like to visit the **SkyTerrace Observation Desk and Museum**

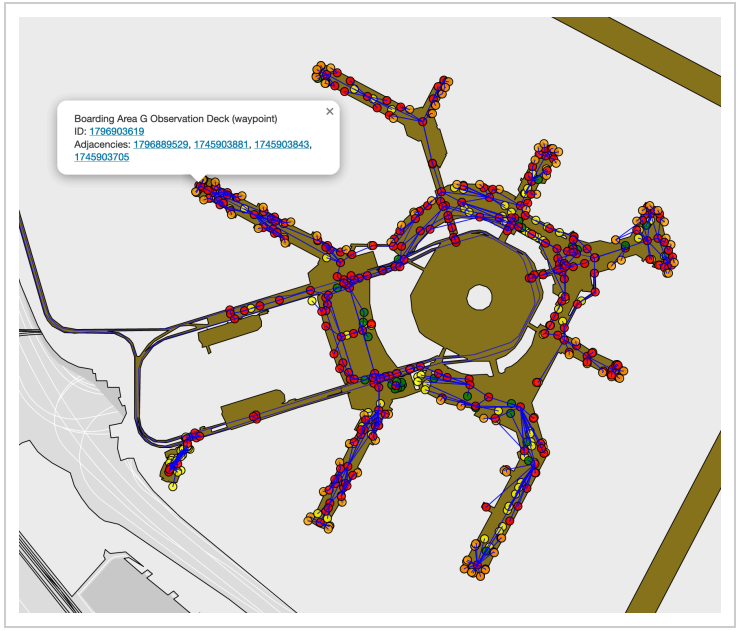
Galleries `https://millsfield.sfomuseum.org/blog/2021/11/19/skyterrace/` in **Terminal 2** `https://millsfield.sfomuseum.org/terminals/1763588123/` :

https://millsfield.sfomuseum.org/wayfinding/#from=1763595139&to=1763588495

People often ask: *Where is the museum at SFO?* It's an understandable question and many people will point to the **SFO Museum galleries** `https://millsfield.sfomuseum.org/galleries/` that happen to be in the terminal they travel to and from most often. Popular answers include the **F-02 North Connection** `https://millsfield.sfomuseum.org/exhibitions/1159160335/` gallery at the top of Boarding Area F and the **2-A Boomerang Gallery** `https://millsfield.sfomuseum.org/exhibitions/1159160649/` in Terminal 2. One or both of these galleries often appear on the maps produced by Apple, Google or OpenStreetMap. Sometimes people mention the **Aviation Museum and Library** `https://millsfield.sfomuseum.org/exhibitions/1159160707/` in the International Terminal. None of these answers are

"wrong" but a more correct way to answer the question (where is SFO Museum) is to say: The airport *is* the museum!

In addition to the dedicated Aviation Museum and Library space, SFO Museum has upwards of **two dozen galleries** `https://millsfield.sfomuseum.org/galleries/` located throughout the terminals as well as over 100 works of art purchased by the **San Francisco Arts Commission** `https://www.sfomuseum.org/public-art` . It is already possible to travel between all the terminals pre-security and you can walk from the end of Boarding Area G, in the International Terminal, all the way to the end of Boarding Area C, in Terminal 1, post-security. Once the final section of the new Harvey Milk Terminal 1 building is complete you'll be able to do the full-loop post-security: From Boarding Area A all the way to Boarding Area G, and everything in between.



A network overview of the waypoints, that are part of the SFO Museum Wayfinding service, in the SFO terminal complex and Grand Hyatt Hotel (February 2023).

That presents anyone visiting SFO with an amazing opportunity to see all the artwork, and objects from the Museum's own **permanent collection** `https://collection.sfomuseum.org` , on display. Until now, though, it hasn't been as easy as it should be to discover what's nearby or, importantly, how to get there from a particular spot in the airport (for example, the gate your flight is leaving from). This new experimental wayfinding service aims to address these issues.

It is considered experimental in the sense that "it works, until it doesn't". It is understood to still contain inaccuracies and may return incorrect results on occasion. There are also user interface and accessibility issues that need to be addressed. Here is a short list of "knowns-knowns":

1. *Locations for some waypoints, particularly public works, are still approximate. These waypoints are closer to where they should be than not but their location on the map may not reflect their exact reality yet.*

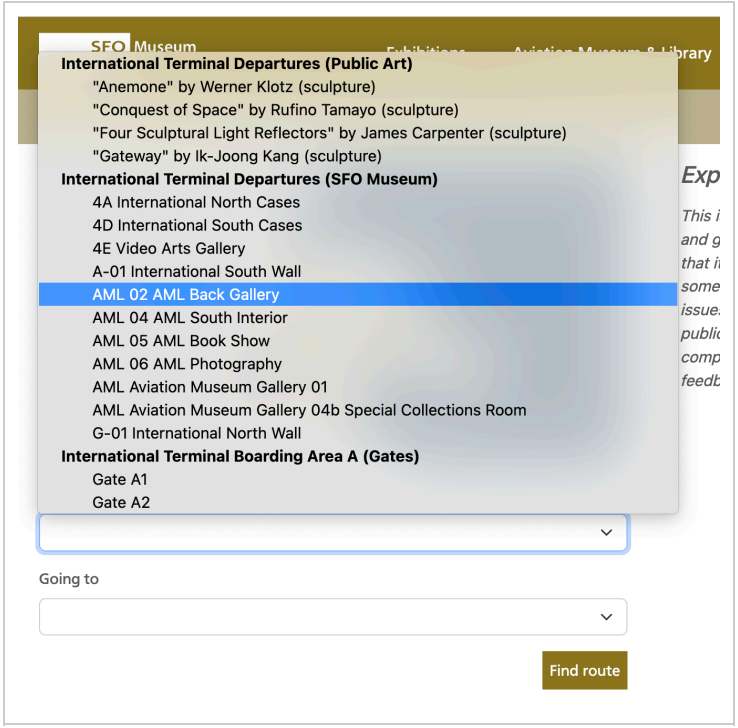
2. *Some routes are missing necessary waypoints in order for that route to make sense. For example not all the doors at SFO have corresponding waypoints which means that, were to you follow routing instructions exactly, you might be directed you to walk through a wall. Please do not try to walk through any walls at SFO.*

3. *Routes do not take in to account (or "weight") whether or not a given path requires going out of, and then back through, TSA security checkpoints.*

4. *Some routes are lacking in the necessary metadata to weight them relative to other options or to exclude them altogether. For example, at least once I've been instructed to travel through the BART station to reach a point in the International Terminal. There's nothing to prevent you from doing this but it will probably cost you a BART fare that you weren't expecting.*

If you're curious to see all the waypoints and their relationship to another have a look at the **Wayfinding Service Network Overview** `https://millsfield.sfomuseum.org/wayfinding/network` webpage. It is also possible to review individual waypoints and their related adjacencies using the **Wayfinding Service Edge Inspection Tool** `https://millsfield.sfomuseum.org/wayfinding/edge` .

What's Next



We are making this experimental service publicly available in the hopes of soliciting feedback and to "learn by doing". We welcome your feedback about this tool. We are especially interested to know if, and where, the tool doesn't work as you expected. You can submit your feedback using **the form on the wayfinding service's about page** `https://millsfield.sfomuseum.org/wayfinding/about#feedback` :

https://millsfield.sfomuseum.org/wayfinding/about#feedback `https://millsfield.sfomuseum.org/wayfinding/about#feedback`

At present, there is no immediate timeline for fixes. Simple things will

be addressed as quickly as possible and harder things will be added to the list so that they aren't forgotten. A short list of improvements we'd like to make as time and circumstance permit include:

1. *Creating new waypoints and updating incorrect waypoints where necessary.*

2. *Integrating the weighting algorithm for determining routes using wait times at the security checkpoints provided by the airport's own* ***Checkpoint API*** `https://developers.flysfo.com/apis/checkpoint-api/index` .

3. *Displaying the names of the exhibitions alongside the gallery they are being shown in, as well as image previews for those exhibitions and public art.*

4. *Improved menus, search and other user-interface elements for choosing and rendering directions between two points at the airport.*

5. *Supporting geolocation and associating those coordinates with their nearest waypoint.*

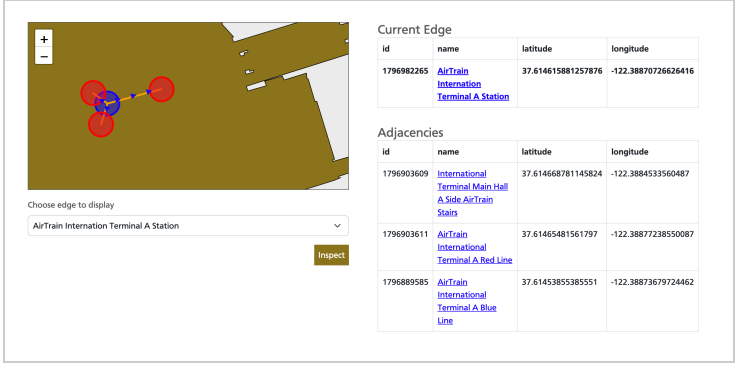
Routing is **a complex subject** `https://valhalla.github.io/valhalla/` and SFO Museum's wayfinding service is relatively primitive by comparison to other wayfinding services. Under the hood it uses a simple directed graph with little to no conditional weighting yet. Some of this is a matter of expediency and some of it by design. The goal of our wayfinding service is not to provide directions precise enough that an automated system, a robot for example, could use them to navigate between terminals. It is meant to be a routing service that provides

sufficient cues to allow a person to get from one part of the airport to another, supplenting the airport's existing signage and an individual's own awareness and curiousity. Airports are busy and dynamic places, after all and if you're not using the giant "directional engine" in your head you are bound to get lost or bump in to something!

The rest of this blog post is about the technical underpinning of the wayfinding service. If you're not interested in that you can stop here and start investigating the wayfinding service itself here:

https://millsfield.sfomuseum.org/wayfinding/ `https://millsfield.sfomuseum.org/wayfinding/`

Under the hood

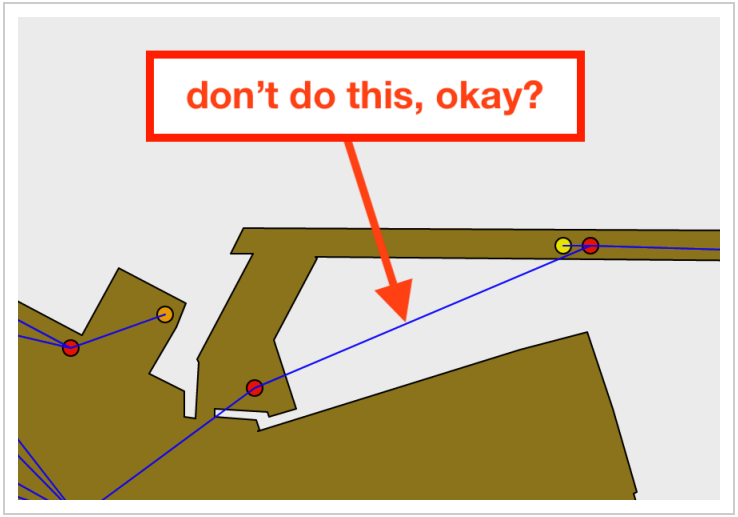


All the data for the waypoints used in the wayfinding service are available as openly-licensed data in the **sfomuseum-data-wayfinding** `https://github.com/sfomuseum-data/sfomuseum-data-wayfinding` GitHub repository. For the most part these are just named points located through the SFO campus with references to their

adjacent waypoints. For example here's a truncated version of the record for the **AirTrain International Terminal A Station** `https://millsfield.sfomuseum.org/wayfinding/edge#edge=1796982265` .


```
{
  "type": "Feature",
  "properties": {
    "millsfield:wayfinding_detail": "ITB",
    "millsfield:wayfinding_kind": "ATR",
    "wof:adjencies": [
      1796903609,
      1796903611,
      1796889585
    ],
    "wof:id": 1796982265,
    "wof:name": "AirTrain Internation Terminal A Station",
    "wof:parent_id": 1763588469,
    "wof:placetype": "venue",
    "wof:placetype_alt": "waypoint",
    "wof:repo": "sfomuseum-data-wayfinding",
  },
  "geometry": {"coordinates":[-122.38870726626416,37.614615881257876],"type":"Point"}
}
```


Generally any two waypoints are rendered as a straight line between one another but sometimes that doesn't make sense. The screenshot below illustrates this point. Although, this particular example has since been addressed (meaning it has an "alternate geometry" described in the next section) it is an example of when drawing a straight line between two adjacent waypoints doesn't make sense.



A straight line suggests that a passenger would travel between the two waypoints by floating in the air outside the airport terminal and over the airfield. Please, don't try to do this. Another example would be drawing a straight line between the International Terminal A and Terminal 2 AirTrain stations (the Terminal 1 AirTrain station is currently closed). Doing so would make it appear as though it were necessary to travel *through* the central parking garage which is not the case. In situations like this a record *may* contain an additional `src:geom_alt` property. This is the case for the record representing the **AirTrain International Terminal A Blue Line stop** `https://static.sfomuseum.org/geojson/1796889585` :

```
"src:geom_alt": [
  "sfomuseum-pathto-1796889549"
]
```

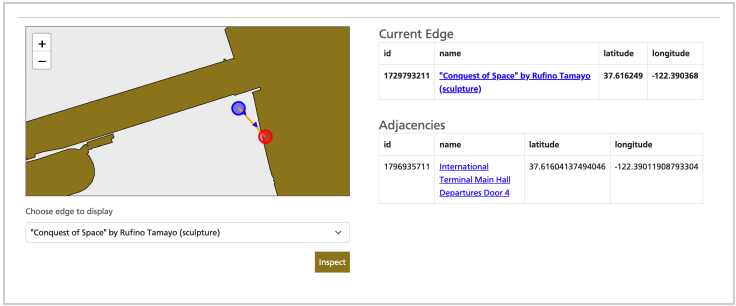
This is a signal to tools consuming the data that the record has **an "alternate geometry"** `https://static.sfomuseum.org/geojson/1796889585-alt-sfomuseum-pathto-1796889549` file; specifically, a file with an

alternate geometry containing the path that the AirTrain follows to travel between the **International Terminal A** `https://static.sfomuseum.org/geojson/1796889585` (1796889585) and **Terminal 2** `https://static.sfomuseum.org/geojson/1796889549` (1796889549) stations. SFO Museum's wayfinding tool checks these signals when rendering waypoints and uses them to render accurate directions.

Waypoints are associated with public art works, SFO Museum galleries and passenger gates at SFO by adding a `millsfield:waypoint_id` property containing one or more waypoints. For example, this is what that property for the record representing Rufino Tamayo's **Conquest of Space** `https://static.sfomuseum.org/geojson/1729793211` , located outside the International Terminal, looks like:

```
"millsfield:waypoint_id": [
```

```
    1796935711
  ]
```



Machine-readable data for public art works as well as galleries and gates are available in the **sfomuseum-data-publicart** `https://github.com/sfomuseum-data/sfomuseum-data-publicart` and **sfomuseum-data-architecture** `https://github.com/sfomuseum-data/sfomuseum-data-architecture` GitHub repositories respectively. Finally, there is a machine-readable endpoint for producing routing directions between any two edges (where an "edge" is the set of all the waypoints and public art works, galleries and gates with `millsfield:waypoint_id` properties) at the airport. For example, here is the request you would make to get directions from Grand Hyatt Hotel Station AirTrain station to Tamayo's Conquest of Space followed by an abbreviated JSON-encoded response:

https://millsfield.sfomuseum.org/wayfinding/api/route?from=1796935709&to=1729793211 `https://millsfield.sfomuseum.org/wayfinding/api/route?from=1796935709&to=1729793211`

```
[
  {
    "id": 1796935709,
    "name": "AirTrain Grand Hyatt Hotel Station",
    "adjacencies": [
```

```
      1796903599,
      1796889551,
      1796903625,
      1729793263
    ],
    "geometry": {
      "type": "Point",
      "coordinates": [
        -122.39332890371237,
        37.613435201247555
      ]
    },
    "is_airtrain": true,
    "airtrain_line": "B",
    "placetype": "waypoint",
    "kind": "ATR",
    "detail": "GHH"
  },

  ...more steps here

    {
    "id": 1729793211,
    "name": "\"Conquest of Space\" by Rufino Tamayo (sculpture)",
    "adjacencies": [
      1796935711
    ],
    "geometry": {
      "type": "Point",
      "coordinates": [
        -122.390368,
        37.616249
      ]
    },
    "level": 3,
    "placetype": "publicart",
    "kind": "ITB",
    "detail": "DEP"
  }
]
```

As of this writing developer-focused documentation for waypoints and wayfinding at SFO Museum is scant and incomplete. For example: What do the `kind` and `detail` properties in the machine-readable response mean? That's not written down anywhere yet (but will be once the dust has settled a bit more). If you have specific questions, requests or suggestions please don't hesitate to file them as new issues in the **sfomuseum-data-wayfinding** `https://github.com/sfomuseum-data/sfomuseum-data-wayfinding` GitHub repository:

https://github.com/sfomuseum-data/sfomuseum-data-wayfinding/issues `https://github.com/sfomuseum-data/sfomuseum-data-wayfinding/issues`

2023-02-14

things I have written about elsewhere #20230216

Using WebAssembly to parse EDTF date strings...

Using WebAssembly to parse EDTF date strings using a Go library in Python

Negative: San Francisco International Airport (SFO), road construction. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.2564***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/02/14/py-edtf` , in February 2023.

This is a short technical blog post to note the release of the **py-edtf** `https://github.com/sfomuseum/py-edtf` Python library for parsing date strings formatted using the Library of Congress' **Extended DateTime Format** `https://www.loc.gov/standards/datetime/` (EDTF),

which was formalized as **part of the ISO-8601 standard** https://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en in 2019. What is notable about this release is that none of the actual code for parsing EDTF strings is happening in native Python code. Instead the `py-edtf` package is invoking code written using the Go programming language and which has been compiled in to a **WebAssembly** https://webassembly.org/ binary and bundled with the rest of its source code.

We wrote about the underlying Go package for parsing EDTF strings in the **Tools for Complex and Ambiguous Dates at SFO Museum** https://millsfield.sfomuseum.org/blog/2021/01/14/edtf/ blog post. We also wrote about invoking that Go code from JavaScript in the **Reverse-Geocoding in Time at SFO Museum** https://millsfield.sfomuseum.org/blog/2021/03/26/spatial/ blog post. Now it's possible to do the same in native Python code. Under the hood, **there's quite a lot going on** # but using the Python library to parse EDTF strings is as easy as:

```
import edtf.parser

p = edtf.parser.Parser()
dt = p.parse("2004-06-XX/2004-07-03")
```

We are releasing this package alongside the **go-edtf** https://github.com/sfomuseum/go-edtf and **go-edtf-wasm** https://github.com/sfomuseum/go-edtf-wasm package (which is used to create the WebAssembly binary bundled with the Python code) in a spirit of generousity. We believe that widespread adoption of EDTF by the cultural heritage sector would be beneficial to all but recognize that it is a complex specification to interpret in code and that not everyone is in a position to support code written in the Go programming language. The `py-edtf` package allows SFO Museum to share the code it has written in Go with others who may

prefer to write their code in Python. This ability to share code across languages using the **WebAssembly binary format** `https://webassembly.org/` is novel because it embodies both the theory and the practice of "small focused tools", by and for the cultural heritage sector, that we've **written about previously** `https://millsfield.sfomuseum.org/blog/2020/08/04/small-tools/` :

> The cultural heritage sector needs as many small, focused tools as it can produce. It needs them in the long-term to finally reach the goal of a common infrastructure that can be employed sector-wide. It needs them in the short-term to develop the skill and the practice required to make those tools successful. We need to learn how to scope the purpose of and our expectations of any single tool so that we can be generous of, and learn from, the inevitable missteps and false starts that will occur along the way.

WebAssembly is a nascent technology so it is still a bit rough around the edges, not uniformly supported across the wide array of programming languages in use and definitely not suitable for certain kinds of complex applications yet. However when it does work, as in the case of parsing and validating EDTF date strings, it points to a whole new set of exciting possibilities.

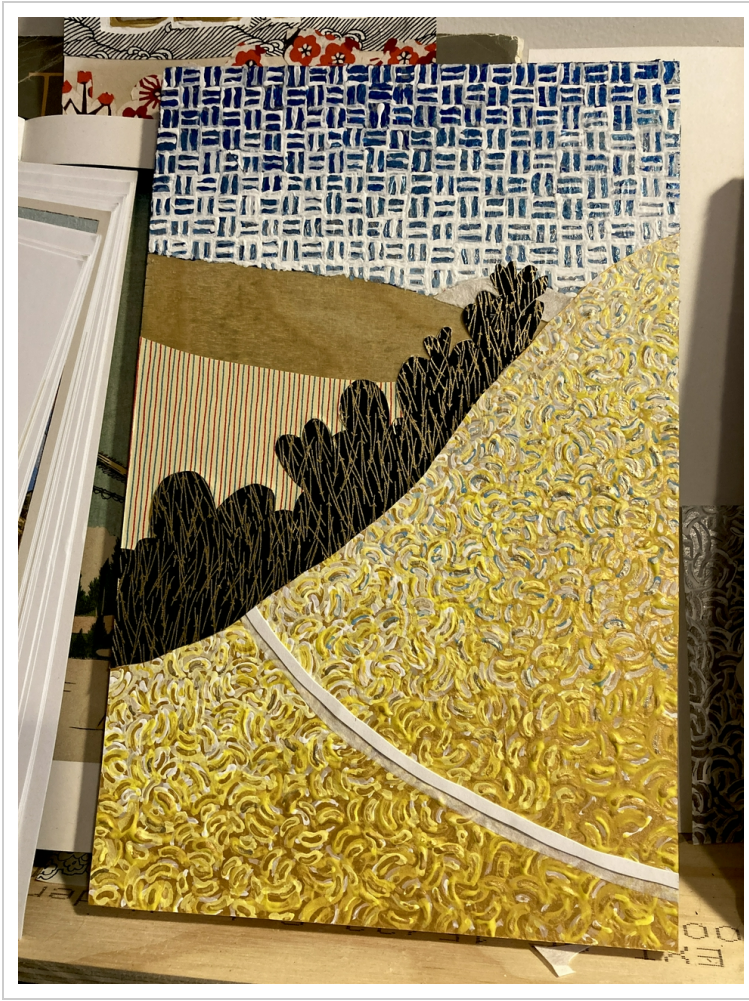
Negative: San Francisco International Airport (SFO), runway construction. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.0520***

2023-02-16

heatmaps of meetings about heatmaps

Dipsea Trail (2022-23)

Dipsea Trail (2022-23)



This one took a while.

This is what it looked like in November. I don't remember when I started it. Maybe September? I thought it was done at this point. I may revisit this still.

And then in January this happened.

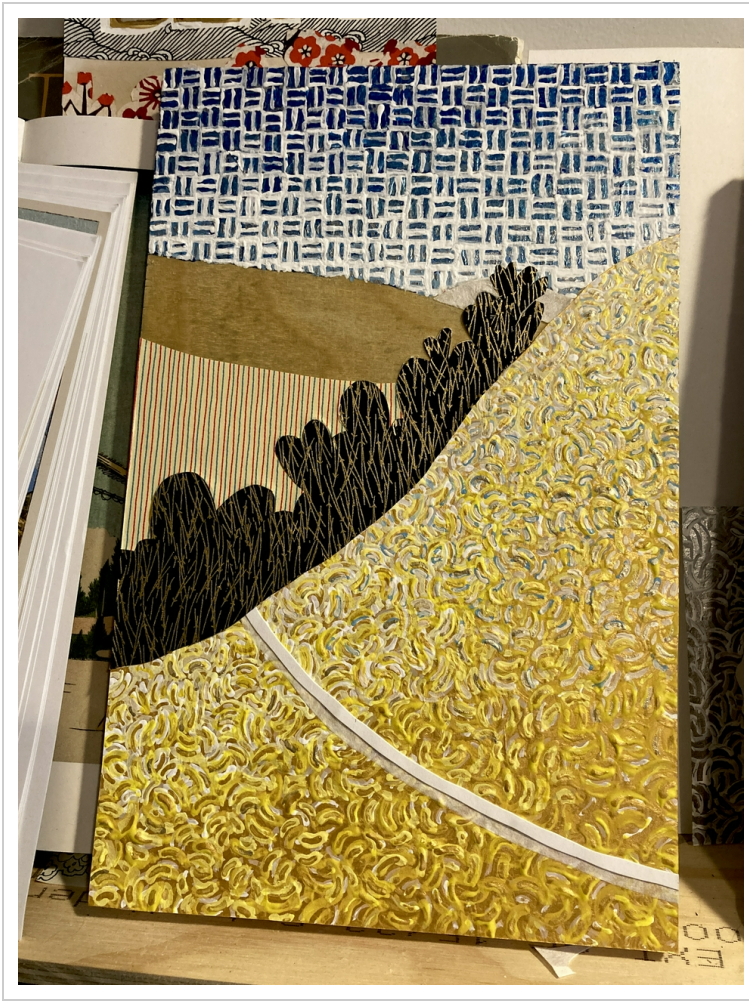
And then this. This was hard to feel happy about.

BEACH

This started to feel better.

Until it didn't. This was a mistake.

Almost done. Maybe?



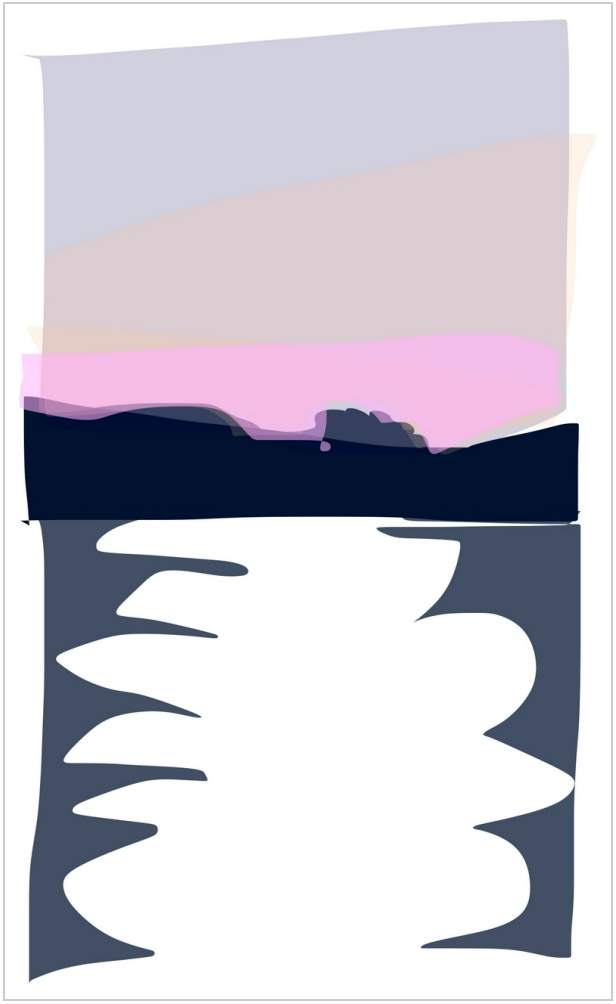
Acrylic and chiyogami and paper on wood. 8 x 13 inches.

2023-02-17

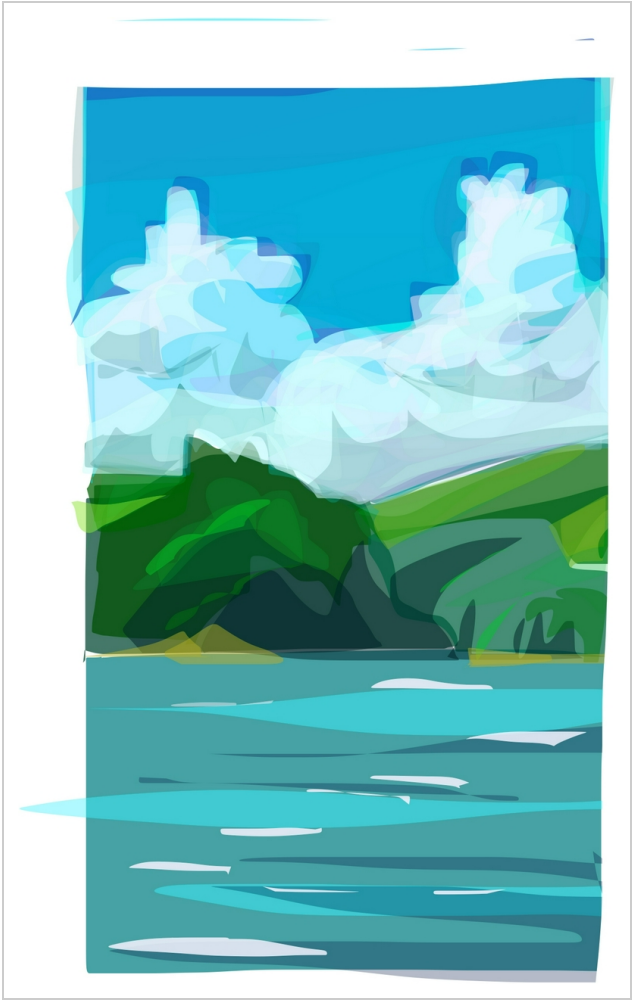
the hills were covered in a seal fur of grass

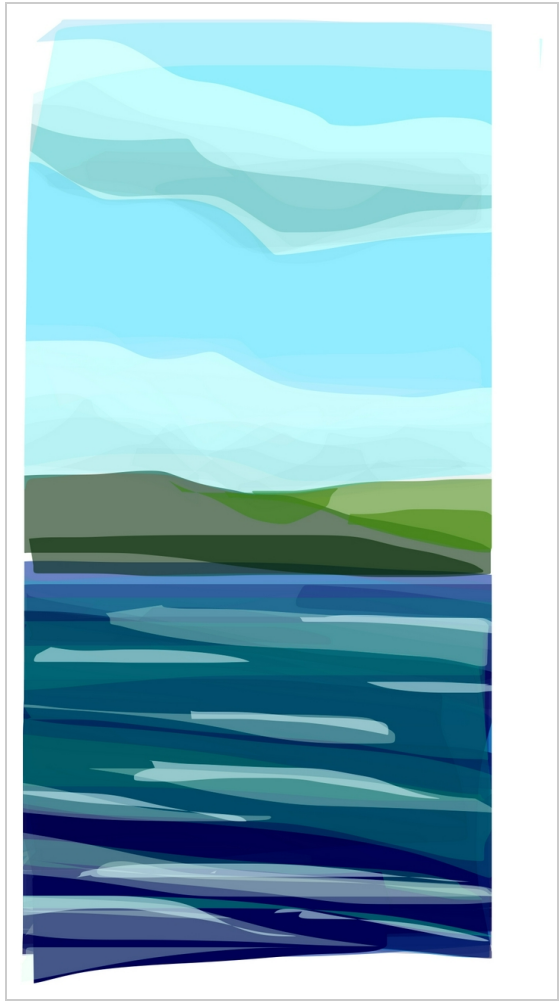
Tomales Bay (2023)

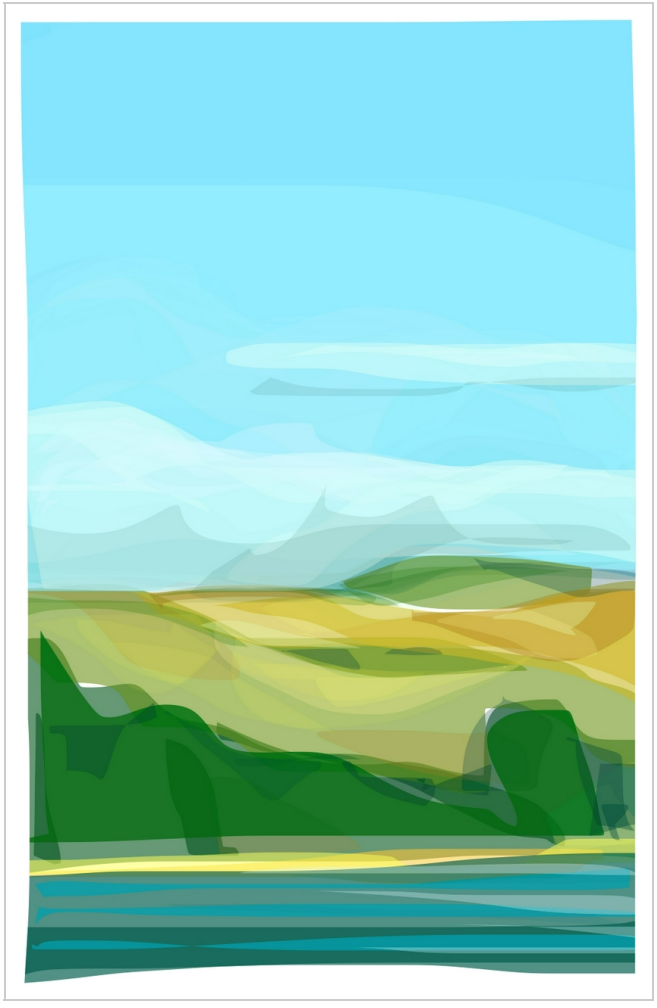
Tomales Bay (2023)

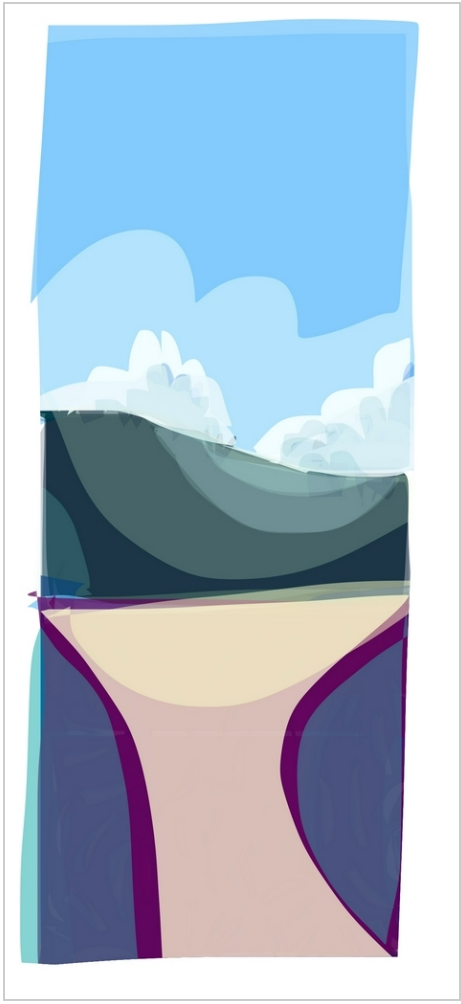


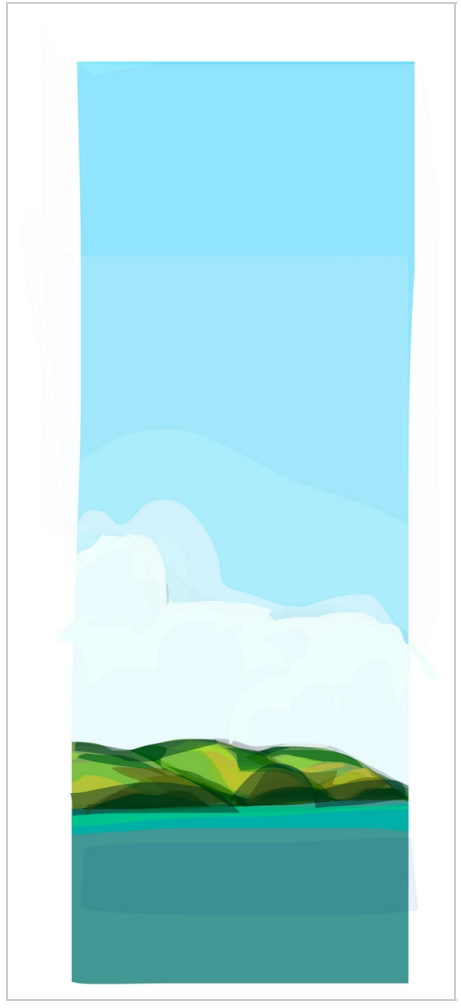
We had the opportunity to spend some time on Tomales Bay looking across the water at Point Reyes recently. These are some of the "drawings" I made on my phone while we were there.

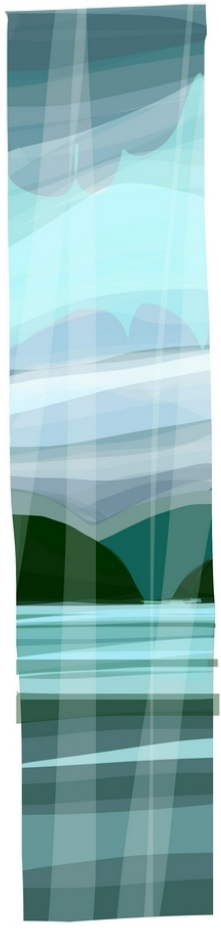




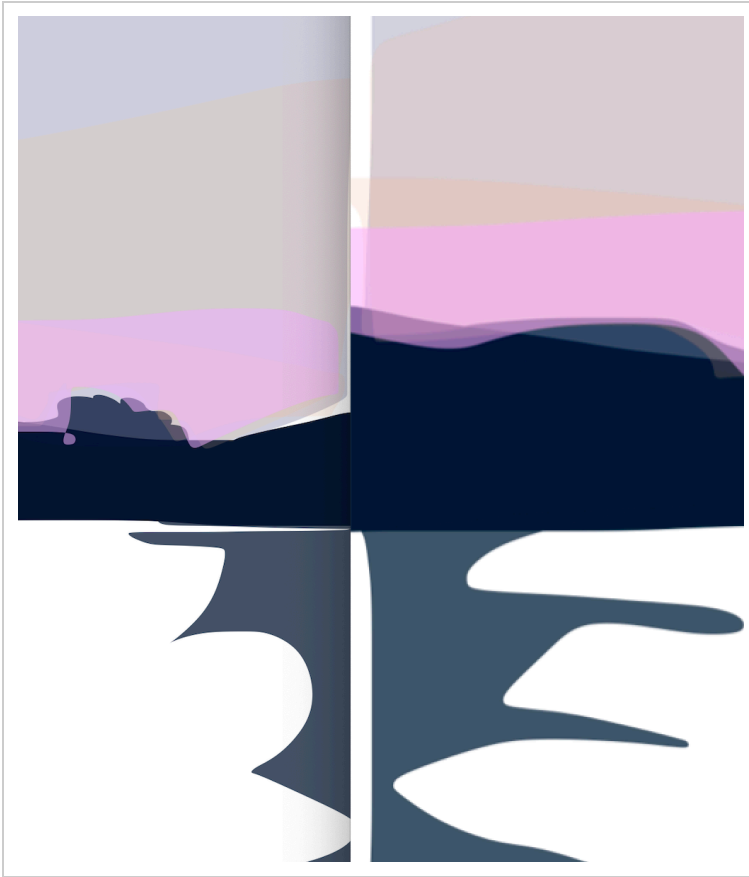








Under the hood



A few years ago I got an iPad and an Apple Pencil to try "drawing" pictures with. I hated it. It's all very impressive technology and there's no question that other people are making some remarkable images with it. It is not my medium, though. Fast forward to last year when Apple started bundling their

Freeform `https://en.wikipedia.org/wiki/Apple_Freeform` shared-whiteboarding application in iOS. Among other things, the application allows you to make freehand drawings using a small set of "tools" one of them being a bezier-fill tool that automatically closes any path you draw and fills it with the selected colour. These images were made this way.

Unlike the Pencil and the many sketching apps which attempt to mimic real-world, analog, tools one of the reasons I find the bezier-fill tool fascinating is because it feels uniquely *digital*. For example, many painters would kill for an analog equivalent and there's a whole other discussion about whether or not this is what Matisse was chasing in his cardboard cut-outs. For me, right now, it's an interesting medium and corresponding set of constraints to work with. It makes me wonder whether being able to adjust the tolerances of those constraints, for example how "taught" a curve is, would be.

If there's documentation on the file format that the Freeform application stores things in I haven't been able to find it. Nor have I been able to find where the files are stored, either on my phone or my laptop. The only way I've found to export these images is as PDF files. Once they are exported as PDF files they are also encoded as raster images rather than the native vector graphics I assume them to be. This is disappointing because in my efforts to figure out how to give these images some degree of permanence outside of a digital infrastructure it would be helpful to have the raw vector instructions. Sol Lewitt jokes are left as an exercise to the reader.

In the meantime I have written **some tools for exporting the Freeform PDF files as high quality JPEG images** `https://github.com/aaronland/go-freeform` with the relevant date information from the former embedded into the latter.

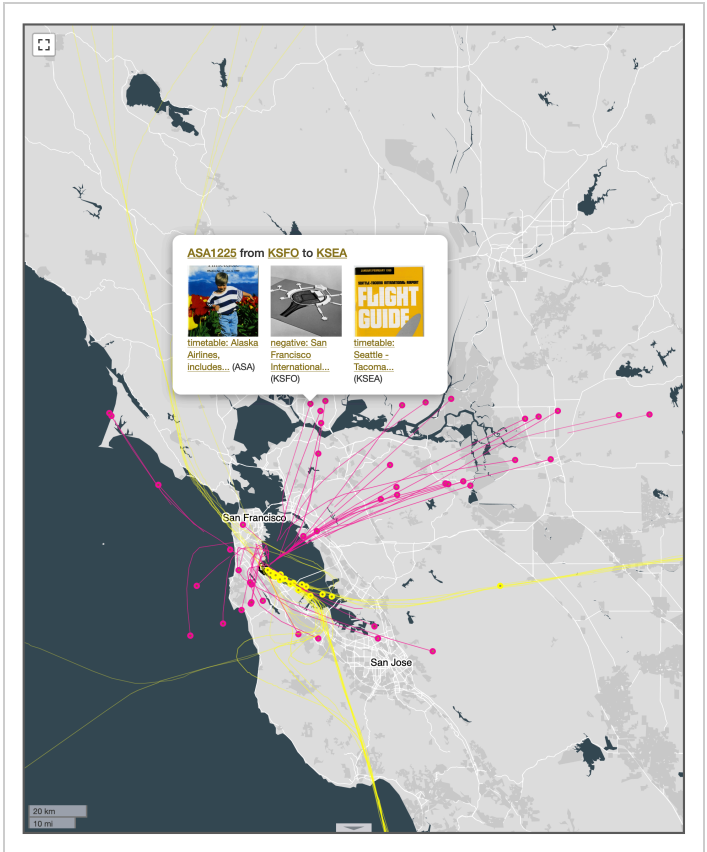
Many thanks to **Mandy Koh's `prism` library** `https://github.com/mandykoh/prism` for wrangling the correct colour space details when translating the PDF files in to JPEG images. This was the source of quite a bit hair-pulling and annoyance before I found Mandy's code. Colour issues continue to plague any attempt to translate these images to print. It's the **same old story** `https://aaronland.info/weblog/2020/10/27/uniformity/#same` and I am still thinking about whether, and how, to approach the question.

2023-03-06

things I have written about elsewhere #20230323

Browsing the SFO Museum Aviation Collection...

Browsing the SFO Museum Aviation Collection Through Real Time Flight Data

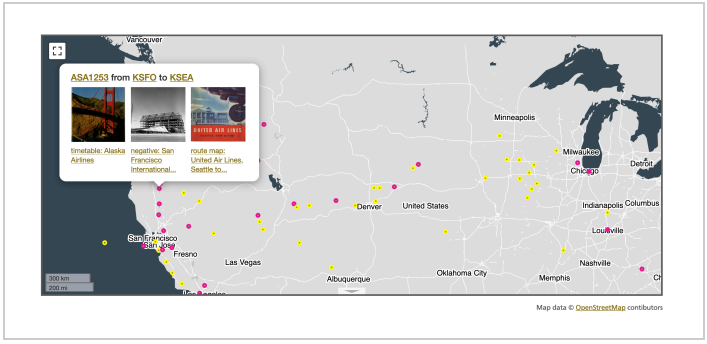


This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/03/23/realtime` , in March 2023.

We are excited to announce a new interface for browsing the SFO Museum Aviation Collection: **Real time (-ish) flights to and from SFO** `https://millsfield.sfomuseum.org/flights/realtime/` . You can try it out on the Mills Field website at:

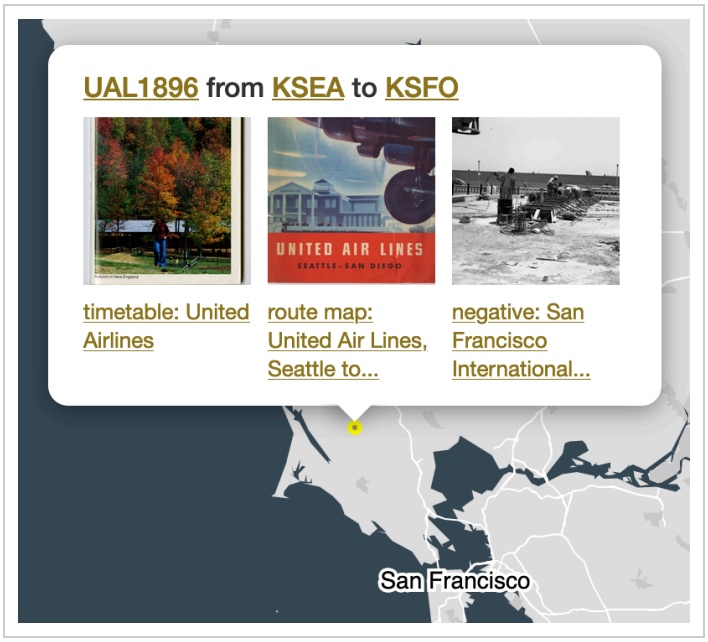
https://millsfield.sfomuseum.org/flights/realtime/ `https://millsfield.sfomuseum.org/flights/realtime/`

I like to say that there is no part of a passenger's time at SFO that shouldn't have a straight line back to the **SFO Museum Aviation Collection** `https://collection.sfomuseum.org/` . Whether it's the **gate** `https://millsfield.sfomuseum.org/gates/` they are flying out of, the **boarding area** `https://millsfield.sfomuseum.org/boardingareas/` or **terminal** `https://millsfield.sfomuseum.org/terminals/` they are waiting for a **flight** `https://millsfield.sfomuseum.org/flights/` in, the **airline** `https://collection.sfomuseum.org/airlines` and **aircraft** `https://collection.sfomuseum.org/aircraft` they are flying on, or even the **airport** `https://collection.sfomuseum.org/airports` they are traveling to, there should be at least one related object from our collection for them to investigate.



With this in mind we've created a map-based interface to display real time (or more specifically real time -ish because there is a delay) flights as they are traveling to and from SFO, inside North America. For each of those flights we know the airline servicing the flight as well as its origin and destination airport. As a flight's position is updated we plot its location on a map and display a popup with a random object for each flight-related

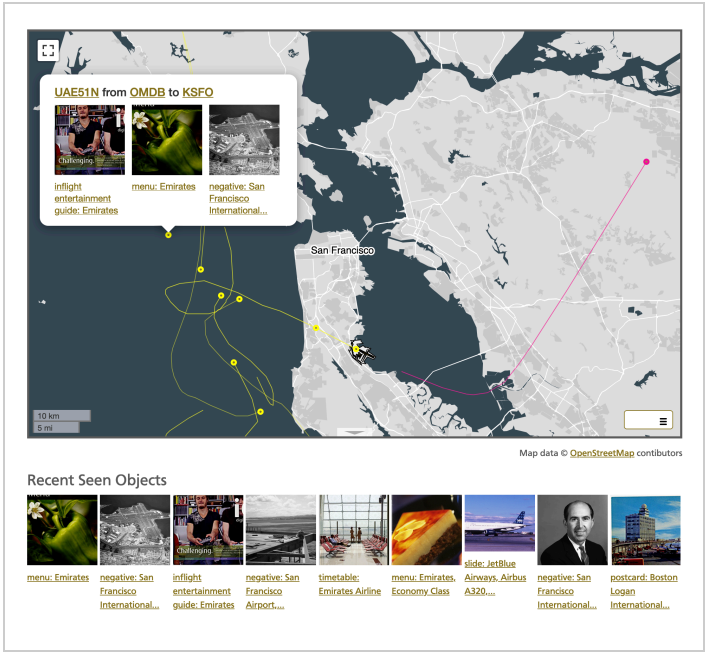
facet from the SFO Museum Aviation Collection. When you click on an object its dedicated web page on the **collection.sfomuseum.org** `https://collection.sfomuseum.org` website will be opened in a new tab.



You can click on any flight to see its popup but as a practical matter popups are only opened automatically when the size of map displayed is less than 1000 miles (a little more than the distance between San Francisco and Denver). If automatic popups are displayed on larger maps and it happens to be a particularly busy moment for travel to and from San Franciso the interface goes a little mad and behaves like this:

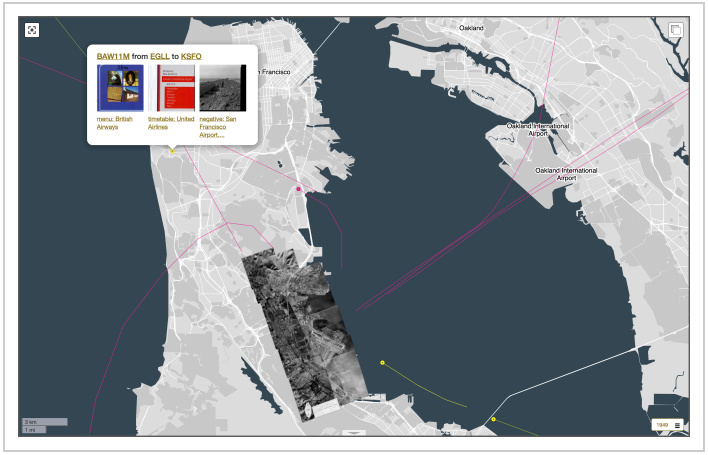
00:00 00:00

Sometimes there is enough flight traffic that there isn't time to see all the objects associated with a flight before a new popup, for another flight, appears. To account for this there is also a carousel of recently seen objects displayed beneath the map. Newer objects are added to the left-hand side of the carousel with older objects being removed as necessary. Each of these objects links to its dedicated web page on the **collection.sfomuseum.org** `https://collection.sfomuseum.org` website.

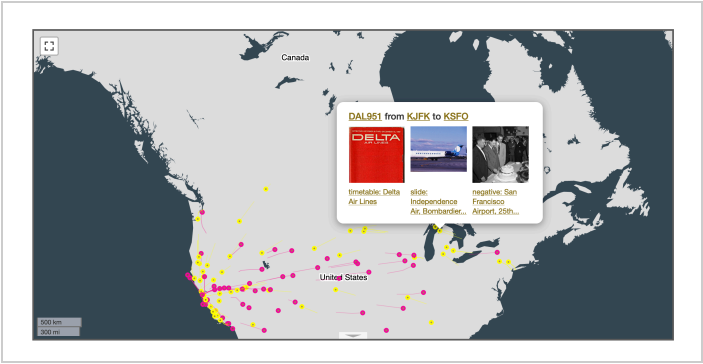


Popups will only be opened automatically for flights that are within the map's current viewport. For example, if you have the map centered on the San Francisco Bay Area then you won't see popups for a flight departing Boston's **Logan Airport** `https://collection.sfomuseum.org/airports/102530741/`. You can disable automatic popups by clicking anywhere on an open popup or by clicking on a different flight to open its popup. Automatic popups will resume once the currently-open popup is closed.

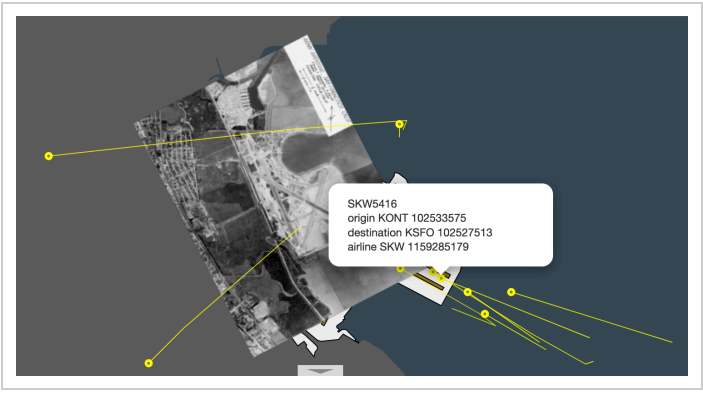
Sometimes we may not be able to derive objects for a given airline or airport, as you can see in the screenshot, above, of a Breeze Airways flight from San Bernadino to SFO. That might be because we haven't collected anything for that subject yet (Breeze only just started **flying in and out of SFO in 2022** `https://www.flysfo.com/media/press-releases/new-us-airline-breeze-announces-plans-serve-four-destinations-sfo`) or because those objects haven't been published to the **SFO Museum Aviation Collection website** `https://collection.sfomuseum.org` yet. We do have *lots* of **objects about SFO** `https://collection.sfomuseum.org/airports/SFO/` , though, so every popup will have a least one object to explore further.



The map itself has controls for displaying **historical aerial maps of SFO** `https://millsfield.sfomuseum.org/map` so you can see how contemporary flight patterns compare relative to the shape and size of SFO throughout the years. Objects in the aviation collection have associations with the "airfield" related concepts they represent (airline, aircraft, airport) but they also have associations with **the place(s) those objects depict or are from** `https://collection.sfomuseum.org/regions/` . A logical next step for this interface is to ask what region a given flight is passing through and then to include a random object from that place alongside all the other objects. There is, after all, a whole country's worth of objects in the SFO Museum Aviation Collection to explore.



Until then here's a screenshot of some early work demonstrating resolving ICAO airline and airport identifiers to their SFO Museum equivalents and a reminder that all of this data, from objects and airlines and airports to individual flights, is published under a liberal use license and available from the **sfomuseum-data GitHub repository** `https://github.com/sfomuseum-data` .



The rest of this blog post is about some of the technical underpinnings of the real time flight map. If you're not interested in that you can stop reading now and start investigating the real time flight map itself here:

https://millsfield.sfomuseum.org/flights/realtime/ `https://millsfield.sfomuseum.org/flights/realtime/`

Under the hood

The real time map fetches collection objects for the airline and airports associated with a flight using an **OEmbed** `https://oembed.com/`

endpoint for random objects. You can use the endpoint to return a random object from the entire **SFO Museum Aviation Collection** `https://collection.sfomuseum.org/` or a random object for a specific airline or airport, using the `?airline_id=` or `?airport_id=` query parameters respectively. For example:

```
$> curl -s 'https://collection.sfomuseum.org/random/oembed?airport_id=102527513' | jq

{
  "version": "1.0",
  "type": "photo",
  "width": 973,
  "height": 767,
  "title": "negative: San Francisco International Airport (SFO), fire department",
  "url": "https://static.sfomuseum.org/media/152/782/407/3/1527824073_dkpiNzWKGuzD8phDsTd
  "author_name": "SFO Museum",
  "author_url": "https://collection.sfomuseum.org/objects/1511945391/",
  "provider_name": "SFO Museum",
  "provider_url": "https://collection.sfomuseum.org/",
  "sfomuseum:accession_number": "2011.032.0959"
}
```

By the time a flight has been relayed to the real time map its airline and airport IDs have already been derived from their **ICAO equivalents** `https://en.wikipedia.org/wiki/International_Civil_Aviation_Organization` . For example, `KSFO` becomes **102527513** `https://collection.sfomuseum.org/airports/102527513/` and `DAL` (the ICAO airline code for Delta Airlines) becomes **1159284261** `https://collection.sfomuseum.org/airlines/1159284261/` and so on. We perform these lookups, internally, using the **sfomuseum/go-sfomuseum-airfield** `https://github.com/sfomuseum/go-sfomuseum-airfield` package, written in Go. In the spirit of investigation and generousity we are releasing a **go-sfomuseum-airfield-wasm** `https://github.com/sfomuseum/go-sfomuseum-airfield-wasm` package which makes that same lookup functionality available as a WebAssembly binary that exposes JavaScript methods for looking up SFO Museum-specific identifiers matching IATA and ICAO codes. For example:

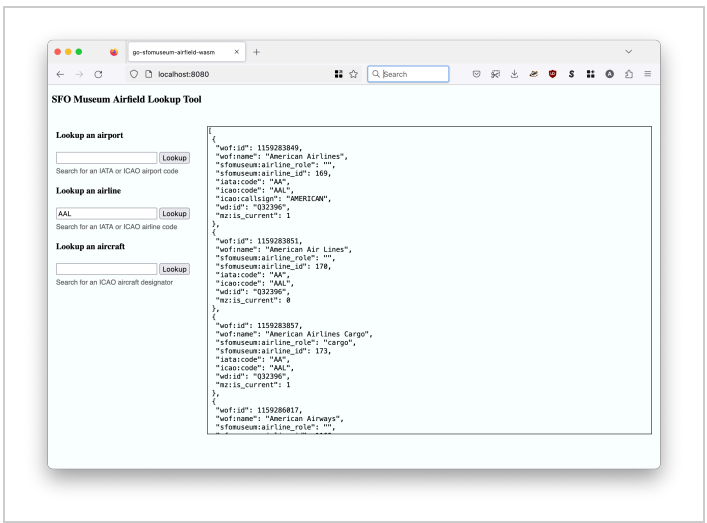
```
sfomuseum_lookup_airport("KSFO")
    .then((data) => { ... });
```

In this example, the airport code `KSFO` returns the following response, as

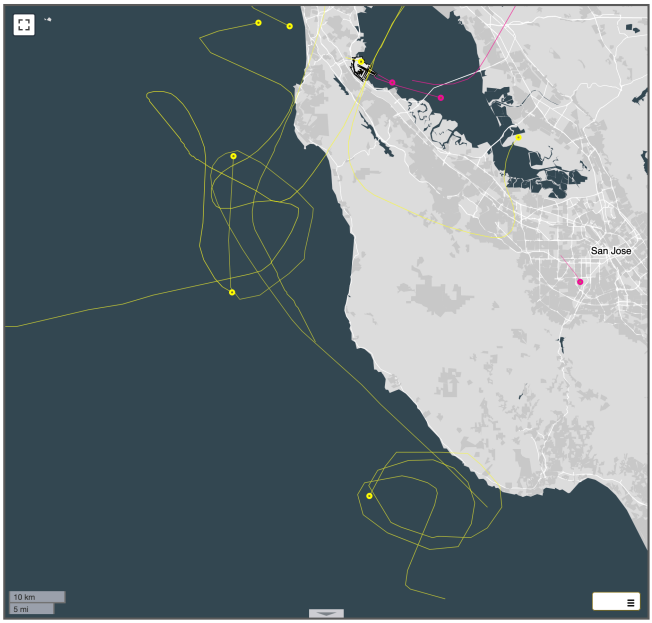
a JSON-encoded string:

```
[
 {
  "wof:id": 102527513,
  "wof:name": "San Francisco International Airport",
  "sfomuseum:airport_id": 121,
  "iata:code": "SFO",
  "icao:code": "KSFO",
  "wd:id": "Q8688",
  "mz:is_current": 1
 }
]
```

Here's a screenshot from the **example** `https://github.com/sfomuseum/go-sfomuseum-airfield-wasm#example` website included with the package, which allows you to query airlines, airports and aircraft (that are associated with the SFO Museum Aviation Collection). See the way multiple different airlines all have the same `IATA` and `ICAO` airline codes? Yeah, tha... but fortunately they all have unique, stable and permanent **Who's On First identifiers** `https://millsfield.sfomuseum.org/blog/2018/12/03/airlines/` .



The WASM binary that is produced by the `go-sfomuseum-airfield-wasm` package is very large, like 11MB large, which means it may be prohibitive for any kind of production, internet-facing use. This is particularly true considering that it is possible to compile a static lookup table and query it directly in JavaScript at a fraction of the size of the WASM binary. These are all "known knowns" but WebAssembly continue to be exciting and useful for **a variety of reasons we've talked about in past blog posts** `https://millsfield.sfomuseum.org/blog/tags/webassembly` so it's worth seeing what it makes possible now and because, eventually, some day the binaries won't be so big. Onwards!



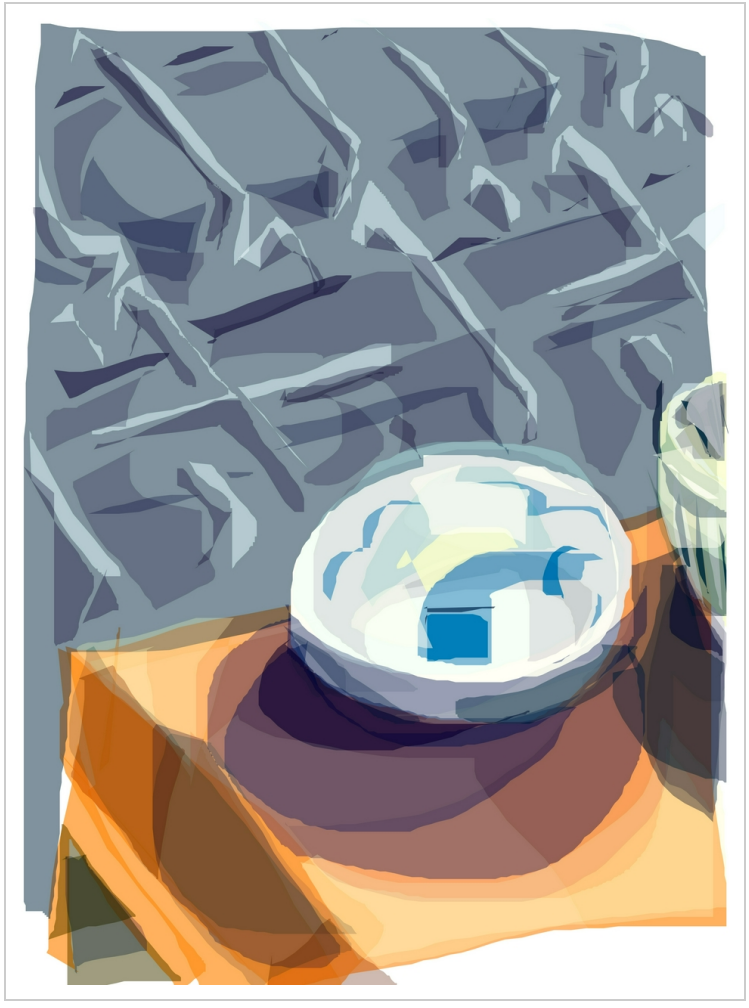
2023-03-23

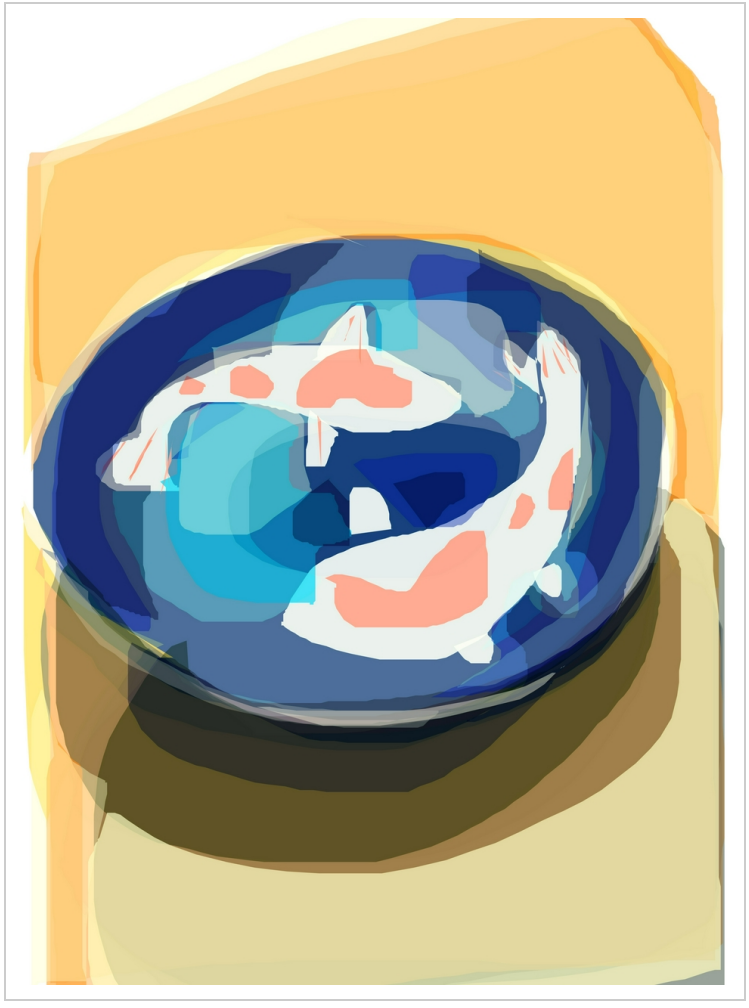
fingerprint

kitchen friends (2023)

kitchen friends (2023)

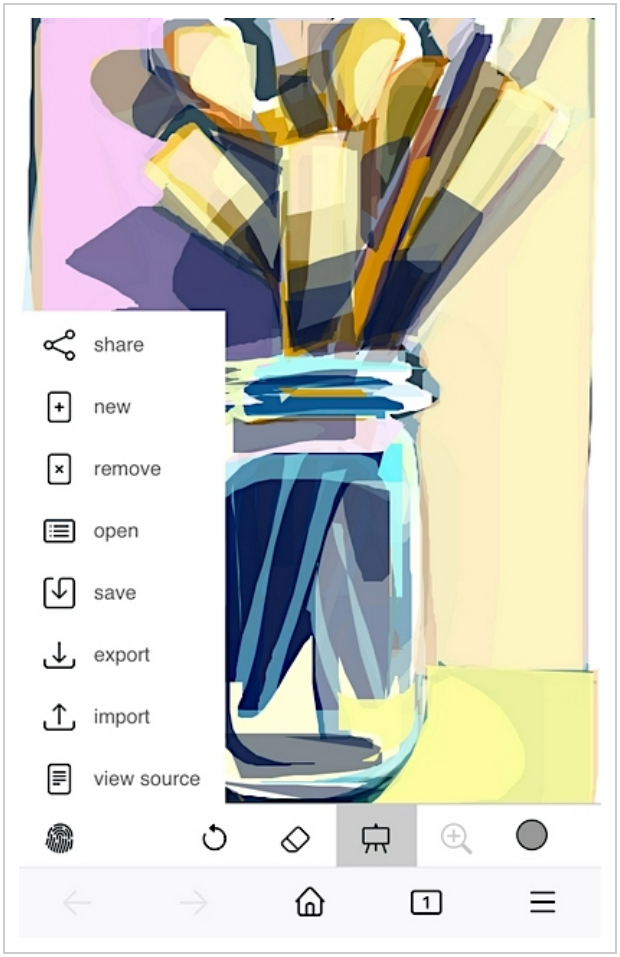
These are some pictures I've made recently of things in, or near, my kitchen.





Some of these are love songs to the work of **Euan Uglow** `https://en.wikipedia.org/wiki/Euan_Uglow` . I continue to be frustrated by the fact that whenever I am able to visit London none of the museums who have collected his works have any of them on display.

Under the hood



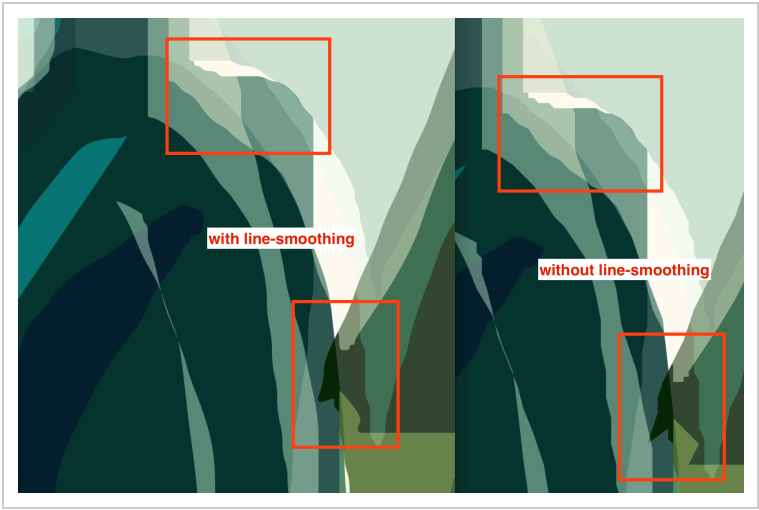
These images were all made with a web-based drawing tool I created called

fingerprint `https://github.com/aaronland/fingerprint/` . In

the last blog post `https://www.aaronland.info/weblog/2023/03/06/fur/#tomalesbay` I mentioned how disappointing it was not to be able to export vector-based representations of drawings made in Apple's Freeform application. `fingerprint` is my way of addressing that disappointiment. It is built on top of Dimitry Baranovskiy's **raphael.js** `https://github.com/DmitryBaranovskiy/raphael` and a modified version of Ian Li's **raphael-sketchpad** `https://github.com/ianli/raphael-sketchpad/` and is the continuation of an earlier drawing tool I made, almost 10 years ago, called **panels** `https://github.com/straup/panels-www` .

It is a single-page HTML5 web application for drawing closed paths (technically strokes that automatically close themselves) filled with a single colour (and opacity). The application has the ability to "undo" and "redo" individual paths as well the ability to remove individual paths by clicking on them (but I almost never use the feature). That's all it does, by design.

`fingerprint` remains a work in progress. It can run offline which is good but it is unable to zoom in and out of an image which is challenging on small devices, particularly when fullscreen web applications are unable to remove all the web browser's chrome (and the valuable screen real estate they occupy) in iOS. Its native file format is little more than a collection of SVG path elements which make rasterizing those images **very easy** `https://github.com/aaronland/go-fingerprint#convert` . It is still a collection of line strings, rather than curves, which means there are frequently "steps" in the drawings.



I have done some initial investigations using **Chaikin's line-smoothing algorithm** `https://github.com/aaronland/go-chaikin` and determined that it is both too slow to perform real-time, in a browser context, and even when applied after the fact doesn't meaninfully change the quality of the line work. So, it's back to the "drawing board" to think about alternative approaches but I am pleased to have a tool whose constraints I understand and which can be improved upon.

2023-04-03

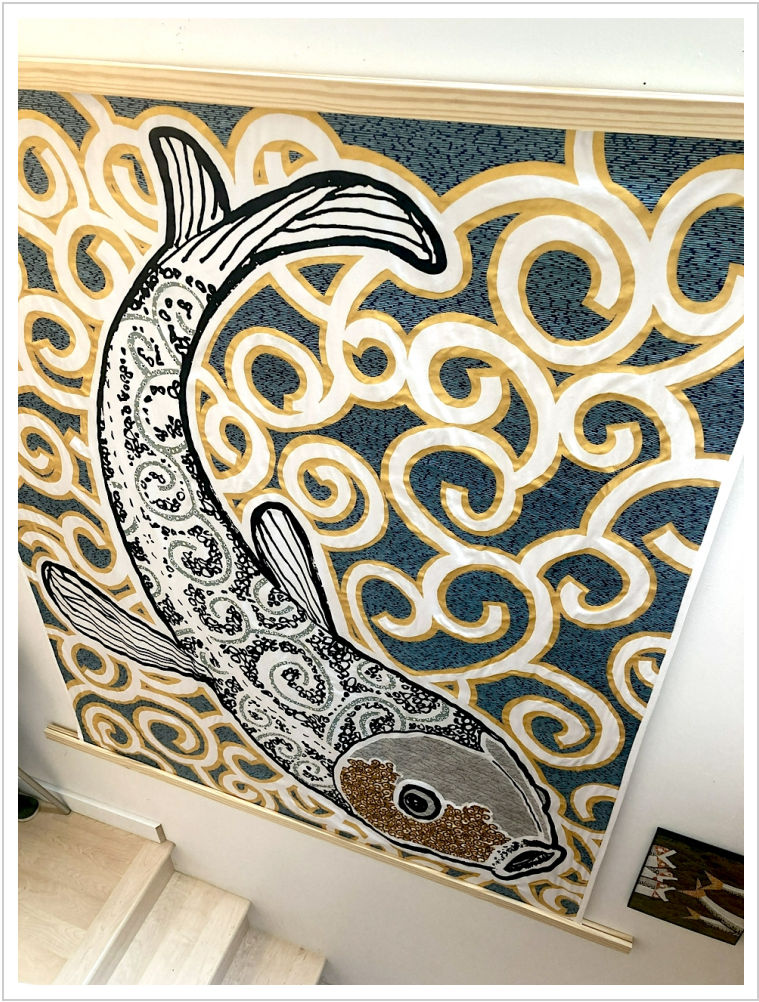
colour profiles for disappointment

big fish (2023)

big fish (2023)

This piece began life as a **4 x 5 inch pen and ink drawing** https://aaronland.info/orthis/171/316/826/5/ in December, 2019. Earlier this year I did the **same-but-different** https://aaronland.info/weblog/2020/10/27/uniformity/#same thing and had it printed on a six-foot length of fabric.

This felt like a good idea until the precise moment when the acrylic gel used to bind the paper to the fabric dried.



56 x 69 inches. Digital print, choyogami, katazome and acrylic on fabric. The smaller piece to the right is 9 x 12 inches, for scale.

Seb Chan https://www.freshandnew.org/ was in town last month for the Museums of Tomorrow Roundtable. He wrote up **the talk that he delivered at the public forum** https://sebchan.medium.com/museums-of-tomorrow-roundtable-2023-words-from-art-museums-and-technological-change-ba56e85be06d , held on the last day of the symposium.

One of the things he mentioned during his talk which didn't make it in to the written version is the idea that visitor's "experience" of the works that museums show is ever-more mediated by the images of those works. This is a longstanding phenomenom, think of the distorted expectations people bring to paintings having only ever seen four-colour (CMYK) reproductions of them in postcards, but one that has been amplified almost beyond understanding with the rise of social media.

I mention all of that because if it weren't hard enough just to finish a piece like this I am constantly filled with dread at the work involved in documenting it. I suppose my life would be easier if I just used someone else's software tools for working with images, notably images captured on a camera phone, but I don't and this morning I discovered that Apple uses **its own colour**

space `https://developer.apple.com/documentation/coregraphics/cgcolorspace/1408916-displayp3` for iPhone images; a colour space not handled by any of my tools yet, which explains why the scaled and resized versions often look dull and muted.

So in addition to the blue light of the sun and the thick yellow light of overhead lights, both of which have been "mediated" by someone else's math in a whisper-tree of software, all of these images have also been juiced by hand to try and give them a little more life than matrix multiplication affords on its own.

2023-05-02

things I have written about elsewhere #20230612

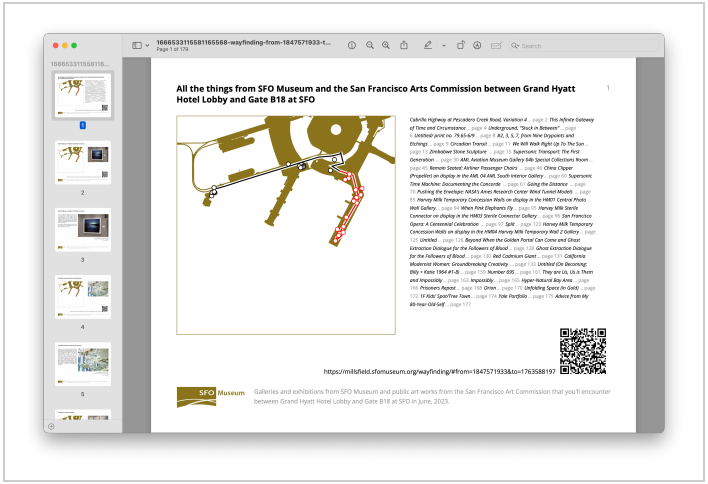
Updating the SFO Museum Wayfinding Service -...

Updating the SFO Museum Wayfinding Service - Publications

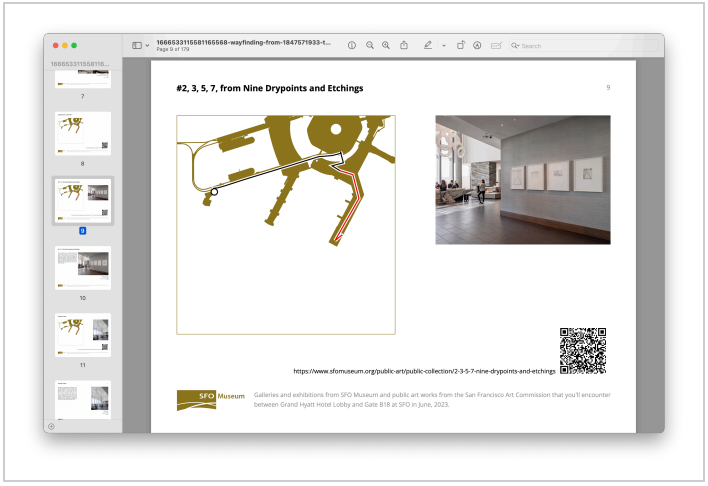
Postcard: American Airlines, Douglas DC-3. Paper, ink. Gift of Thomas G. Dragges, SFO Museum Collection. ***2015.166.0232***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/06/12/wayfinding-pdf/` , in June 2023.

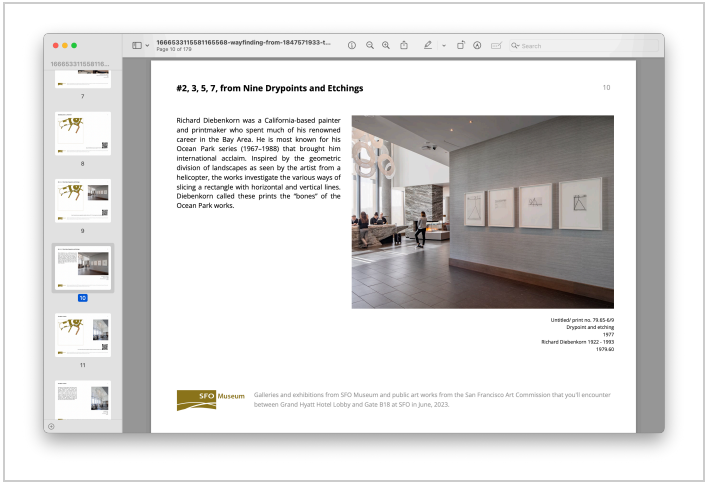
We are pleased to announce the launch of printable "publications" for the **SFO Museum Wayfinding Service** `https://millsfield.sfomuseum.org/wayfinding/blog/tags/wayfinding` . These publications are PDF files depicting all the SFO Museum galleries and exhibitions and public art works from the San Francisco Arts Commission on display along a given route at SFO.



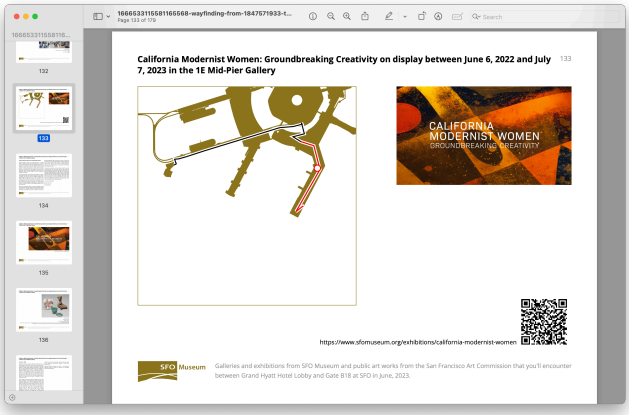
Each publication starts with a "table of contents" page showing a map of the route and all the exhibitions, galleries and public art works along that route. The name for each exhibition and publication art work is listed alongside the page number where photos and text about each object are included. There's also a QR code at the bottom of the page which links to the corresponding wayfinding route on the **Mills Field website** `https://millsfield.sfomuseum.org/wayfinding/` .

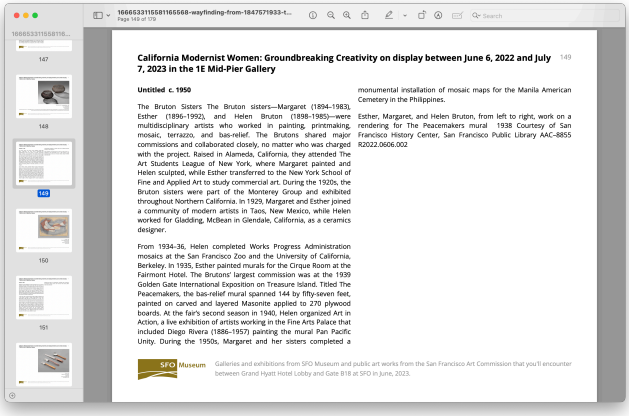


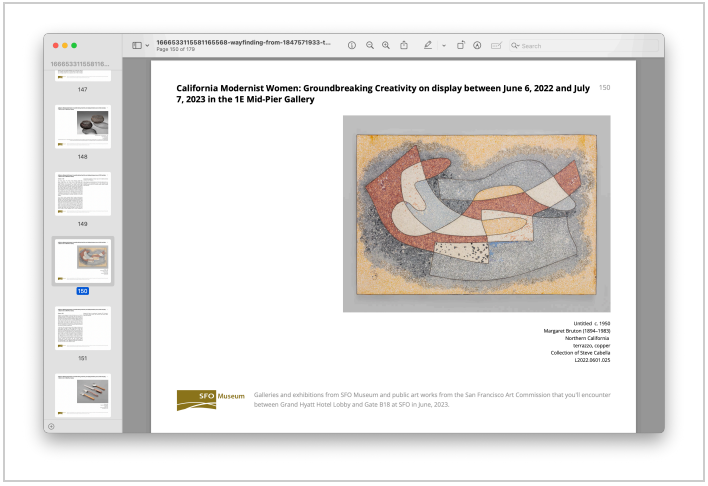
Every exhibition and each public art work has its own "title" page that follows the same model: A map is displayed on the right with the complete wayfinding route and a marker for that work along the route. A small image and the title of the exhibition or object is displayed on the left-hand side of the page and there is a QR linking to that exhibition or object on the **sfomuseum.org** `https://sfomuseum.org/` website below.



Public art works are followed by a single page containing a larger image, a descriptive text and credit line information. Exhibition objects are displayed using a two-page layout for each object. The first page contains a title and descriptive text. The second page displays the title, a large image and credit line information. The texts for individual objects are often longer than the space afforded on a page that also contains an image so we've split them in two.

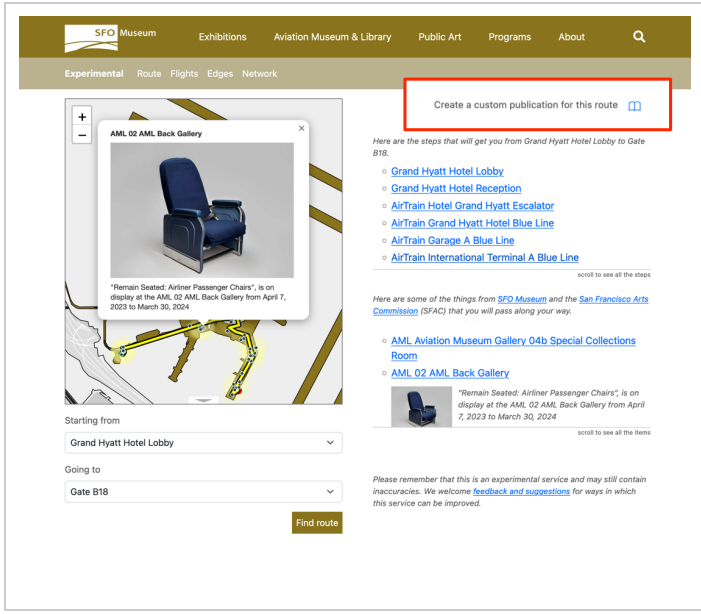

1666533115581165568-wayfinding-from-1847571933-t...
Page 133 of 179
Search
132
133
134
135
136
California Modernist Women: Groundbreaking Creativity on display between June 6, 2022 and July 7, 2023 in the 1E Mid-Pier Gallery
133
CALIFORNIA
MODERNIST WOMEN
GROUNDBREAKING CREATIVITY
https://www.sfomuseum.org/exhibitions/california-modernist-women
SFO Museum
Galleries and exhibitions from SFO Museum and public art works from the San Francisco Art Commission that you'll encounter between Grand Hyatt Hotel Lobby and Gate B18 at SFO in June, 2023.


1666533115581165568-wayfinding-from-1847571933-t...
Page 149 of 179
Search
147
148
149
150
151
California Modernist Women: Groundbreaking Creativity on display between June 6, 2022 and July 7, 2023 in the 1E Mid-Pier Gallery
149
Untitled c. 1950
The Bruton Sisters The Bruton sisters—Margaret (1894–1983), Esther (1896–1992), and Helen Bruton (1898–1985)—were multidisciplinary artists who worked in painting, printmaking, mosaic, terrazzo, and bas-relief. The Brutons shared major commissions and collaborated closely, no matter who was charged with the project. Raised in Alameda, California, they attended The Art Students League of New York, where Margaret painted and Helen sculpted, while Esther transferred to the New York School of Fine and Applied Art to study commercial art. During the 1920s, the Bruton sisters were part of the Monterey Group and exhibited throughout Northern California. In 1929, Margaret and Esther joined a community of modern artists in Taos, New Mexico, while Helen worked for Gladding, McBean in Glendale, California, as a ceramics designer.
From 1934–36, Helen completed Works Progress Administration mosaics at the San Francisco Zoo and the University of California, Berkeley. In 1935, Esther painted murals for the Cirque Room at the Fairmont Hotel. The Brutons' largest commission was at the 1939 Golden Gate International Exposition on Treasure Island. Titled The Peacemakers, the bas-relief mural spanned 144 by fifty-seven feet, painted on carved and layered Masonite applied to 270 plywood boards. At the fair's second season in 1940, Helen organized Art in Action, a live exhibition of artists working in the Fine Arts Palace that included Diego Rivera (1886–1957) painting the mural Pan Pacific Unity. During the 1950s, Margaret and her sisters completed a monumental installation of mosaic maps for the Manila American Cemetery in the Philippines.
Esther, Margaret, and Helen Bruton, from left to right, work on a rendering for The Peacemakers mural 1938 Courtesy of San Francisco History Center, San Francisco Public Library AAC–8855 R2022.0606.002
SFO Museum
Galleries and exhibitions from SFO Museum and public art works from the San Francisco Art Commission that you'll encounter between Grand Hyatt Hotel Lobby and Gate B18 at SFO in June, 2023.

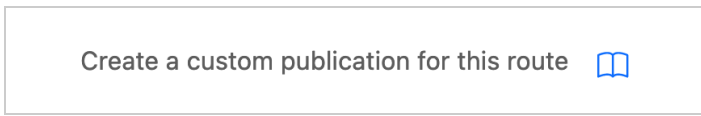


The last page of every publication is a repeat of the map depicting all the things along a route and a QR code the **online feedback page** `https://millsfield.sfomuseum.org/wayfinding/about/#feedback` on the SFO Museum Wayfinding website. In the coming weeks and months we plan to add support for additional formats like the ePub standard used for electronic books. That is further discussed later in this blog post.

How does it work?



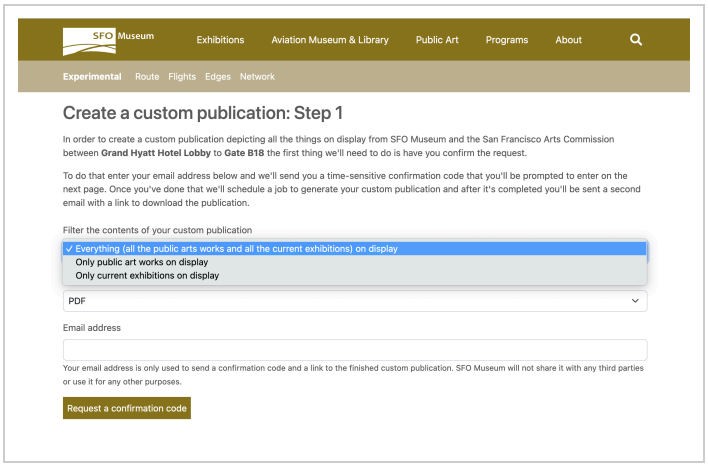
Once you've selected a route a new icon and link to create a custom publication for that route will appear in the top right-hand side of the page.



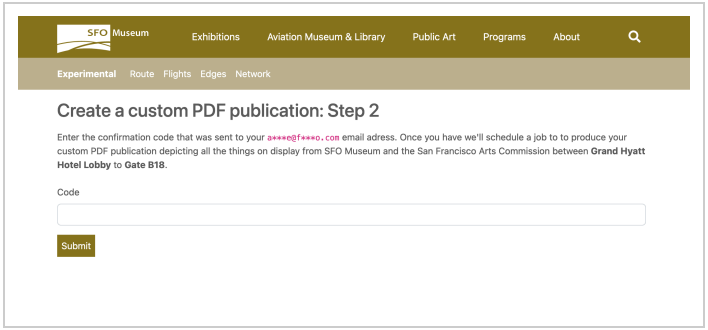
For example, here's what the link to create a custom publication for all the things along the route **between the Grand Hyatt Hotel Lobby to Gate B18** `https://millsfield.sfomuseum.org/wayfinding/#from=1847571933&to=1763588197` looks like:

ld.sfomuseum.org/wayfinding/publication/?933&to=1763588197 `https://millsfield.sfomuseum.org/wayfinding/]33&to=1763588197`

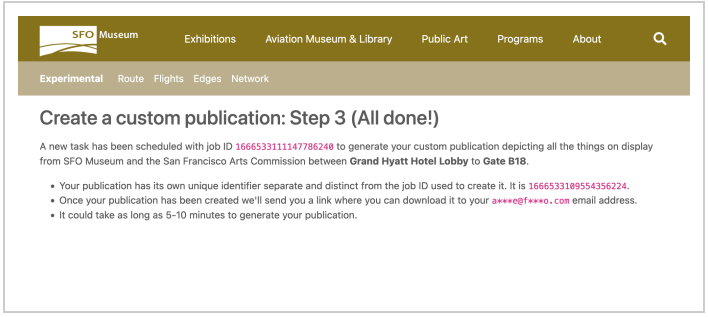
From this page, you can choose whether or not to limit your custom publication to only include exhibitions from SFO Museum or public art works from the San Francisco Arts Commission. The default choice is to include both.



You'll also be prompted to enter an email address which is used to send a time-sensitive confirmation code for your custom publication. *Your email address is only used to send a confirmation code and a link to the finished publication. SFO Museum will not share it with any third parties or use it for any other purposes.*



Once you receive the confirmation code and have entered it in to the form we'll schedule a new task to generate your publication and send you a separate email once it's complete. This second email will contain a link to your publication that you can download and save to your computer. Note that these links are also time-sensitive and will expire in 24 hours.



Why are we doing this?

Employee t-shirt: Virgin America. Cotton, ink. Gift of Lisa Larsen, SFO Museum Collection. ***2018.071.026***

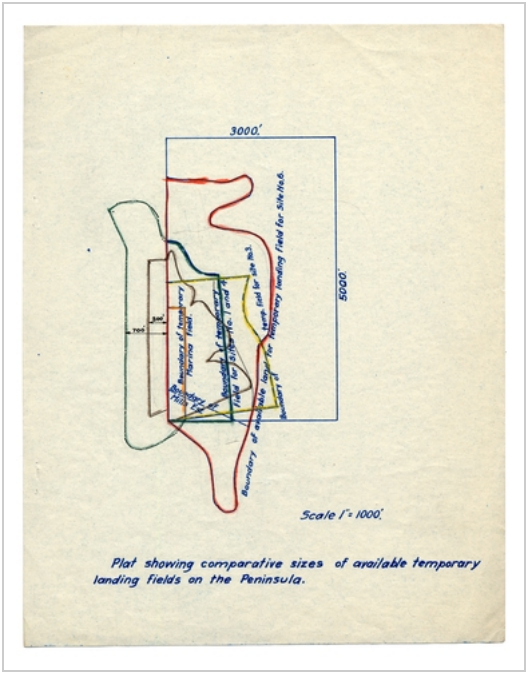
In the **Updating the SFO Museum Wayfinding Service - Galleries, Public Art, Flights and Barcodes** `https://millsfield.sfomuseum.org/blog/2023/05/08/wayfinding/` blog post I wrote:

> Eventually, we can use that same information to compile a PDF file, or a video or even, eventually, an audio guide or podcast about all the work, past and present, that a passenger has passed which they can take with them on to their flight. Not enough museums follow their visitors "out of the building" or offer them the means to continue to think about what they've seen after the fact. This remains the great untapped opportunity for all museums but especially a museum in an airport since, in many ways, "leaving the building" is what an airport is all about.

Custom publications are a first attempt at designing and producing artifacts which can be thought to "follow a visitor out of the building" (or the museum (which also happens to be the airport)). Think of them as training wheels towards achieving that goal. We believe that the objects and public art works on display at SFO Museum and SFO are worth seeing in person but that doesn't, and shouldn't, preclude the ability to enjoy these things when you are not at SFO.

You might choose to create a custom publication to read on your flight after you've gotten to your gate and are waiting to board your plane. You might choose to create a custom publication that you can print out to explore public art works and exhibitions pre-security even if you're not flying out of SFO. You might choose to create a custom publication for a friend who has a layover at SFO and isn't sure what to do. Remember that you can already travel **from one end of Boarding Area G to the other end of Boarding Area C** `https://millsfield.sfomuseum.org/wayfinding/#from=1763588467&to=1763588255` and **from one end of Boarding Area A to the other end of Boarding Area B** `https://millsfield.sfomuseum.org/wayfinding/#from=1763588427&to=1763588217`, post-security, at SFO. Once the new Harvey Milk Terminal 1 building is complete (sometime in 2024) you'll be able to do the "full loop" of all the terminals post-security. That's a whole lot of opportunity to see whole lot of art at SFO (not to mention everything in the **SFO Museum Aviation Collection** `https://collection.sfomuseum.org/` and these publications can be your custom guide along the way.

A note about formats



Plat: San Francisco Department of Public Works, Bureau of Engineering, proposed airport site. Paper, ink. Transfer, SFO Museum Collection. ***2000.018.079.009***

The initial launch of publications only supports one format: PDF files. In the coming weeks we plan to add support for another:
EPUB `https://en.wikipedia.org/wiki/EPUB` , a commonly used format for e-books. PDF files have broad support in all mobile and desktop computers and are familiar to most people. It's also well-suited to printing since that is the heritage the format springs from. Some people have criticized PDF files because they are a less open format than others (like EPUB) and because they typically are designed for letter-sized pieces of paper that may not display well on mobile devices.

This is one of the situations where everyone is right but we had to start somewhere and we chose to start with PDF files. The goal for these publications is to support as many different formats as people desire (and is practical to produce). Creating publications as e-books, which enjoy

better display options for variable-sized devices, is coming soon and if there are other formats you think we should consider **we'd love to hear about them** https://millsfield.sfomuseum.org/wayfinding/about/#feedback .

The rest of this blog post gets in to some technical details about how we've implemented publications for the wayfinding service. If that's not interesting to you this is a good place to stop reading and go try creating your own custom publication.

ield.sfomuseum.org/wayfinding/ https://millsfield.sfomuseum.org/wa

Of if you'd just like to see the example publication, of all the exhibitions and public art works **on display between the Grant Hyatt Hotel Lobby and Gate B18** https://millsfield.sfomuseum.org/wayfinding/#from=1847571933&to=1763588197 , shown in this blog post you can download it here:

'2023/06/12/wayfinding-ublication_20230612_4SBhZJT82sQ.pdf https://millsfield.sfomuseu
ublication_20230612_4SBhZJT82sQ.pdf

Note that this PDF file is 35MB and 179 pages long (reflecting all the work on display at SFO at any given moment). Okay, on to the technical bits.

Under the hood

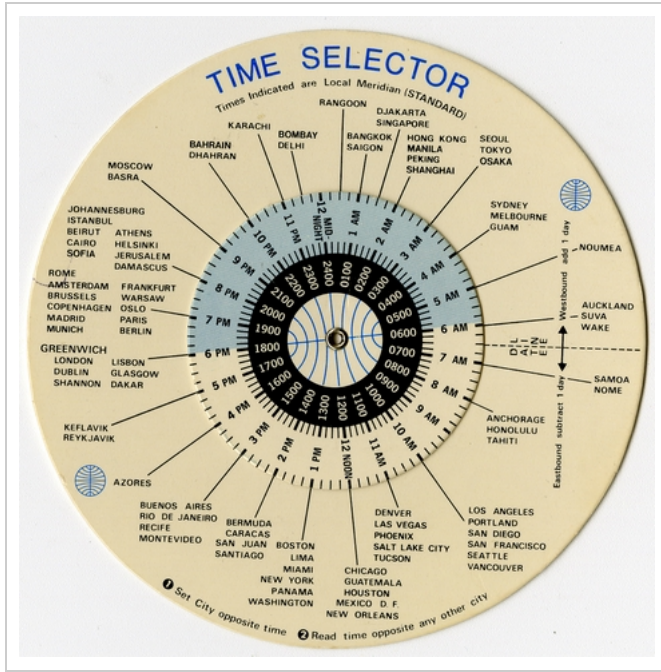
Photograph: Lockheed 18 Lodestar, Fairbanks, Alaska. Photograph. Gift of Philip Smith, SFO Museum Collection. ***2021.069.168***

In addition to the code that generates PDF publications for a wayfinding route we've also developed discrete components for storing and processing offline tasks (like producing a publication), generating, delivering and validating access tokens (for the confirmation codes) and localizing dates and times in messages (in order to tell a person when an access token will expire in the timezone where they are located). These were all developed in the spirit of doing the extra work to produce and share **small focused tools** `https://millsfield.sfomuseum.org/blog/2020/08/04/small-tools/` by and for the cultural heritage sector.

> The cultural heritage sector needs as many small, focused tools as it can produce. It needs them in the long-term to finally reach the goal of a common infrastructure that can be employed sector-wide. It needs them in the short-term to develop the skill and the practice required to make those tools successful. We need to learn how to scope the purpose of and our expectations of any single tool so that we can be generous of, and learn from, the inevitable missteps and false starts that will occur along the way.

We are committed to open sourcing the first two components but they are still a bit rough around the edges and haven't been properly documented yet. We have released the last component, though.

Localizing dates and times



Time converter: Pan American World Airways. Cardboard, ink, metal. Gift of the Pan Am Association, SFO Museum Collection. ***2000.058.1498***

We want to be able to localize dates and times for people visiting SFO Museum websites and in order to do that we need to know "where" they are. Not precisely where they are but just what timezone they are in. To accomplish this we are using **the openly-licensed MaxMind GeoLite2City IP lookup database** `https://dev.maxmind.com/geoip/geolite2-free-geolocation-data/#license` which is bundled as an embedded file in the **sfomuseum/go-geoip2** `https://github.com/sfomuseum/go-geoip2` package and used to derive the timezone associated with the IP address a person has requested a custom publication from.

Embedding the MaxMind data produces some abnormally large binary tools but these file sizes are understood to be an acceptable trade-off. The reason we don't mind the large file size is that we deploy the service to

derive a timezone for an IP address as a simple **AWS Lambda Function URL** `https://docs.aws.amazon.com/lambda/latest/dg/lambda-urls.html` where, aside from the time it takes to upload the service in to AWS, its file size is sort of irrelevant. Deploying the service as an AWS Lambda Function URL with an embedded database means we don't need to think about configuring our service with dedicated filesystems, databases or containers (to manage filesystems or databases).

Finally an AWS Lambda Function URL deployment has the attractive quality of "scaling to zero" so we are only charged when the function is invoked (like when we need to derive a person's timezone from their IP address) rather than running a dedicated server 24 hours a day. This was previously possible using the plain AWS Lambda function paired with an AWS API Gateway pattern but what Function URLs allow is removing the API Gateway piece from the puzzle so that there are fewer moving parts to keep track of.

Here's an example of what that IP address to timezone service looks like run as a web service locally on my computer:

```
$> curl 'http://localhost:8080/timezone?address=219.3.47.6'
Asia/Tokyo
```

And here's what it looks like running as a AWS Lambda Function URL deployment:

```
$> curl 'https://{FUNCTION_URL_ID}.lambda-url.{AWS_REGION}.on.aws?address=219.3.47.6'
Asia/Tokyo
```

This is not all the service *could* do, since it has access to all the interesting data that the MaxMind databases provide, but it is all the service does today. It is very much a "small focused tool", a fast and cheap solution to a specific problem that can be plugged in to a larger purpose-fit application. It works for us and we hope it might be useful for you too.

https://github.com/sfomuseum/go-geoip2 `https://github.com/sfomuseum/go-geoip2`

2023-06-12

things I have written about elsewhere #20230703

Updating the SFO Museum Wayfinding Service -...

Updating the SFO Museum Wayfinding Service - Interface changes, WebShare and Nearby

Towelette: Air France. Plastic. Anonymous gift, SFO Museum Collection. ***2018.028.024***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/07/03/wayfinding-nearby/` , in July 2023.

Since the last blog post about adding **custom publications for wayfinding routes** `https://millsfield.sfomuseum.org/blog/2023/06/12/wayfinding-pdf/` we've made the following changes and improvements:

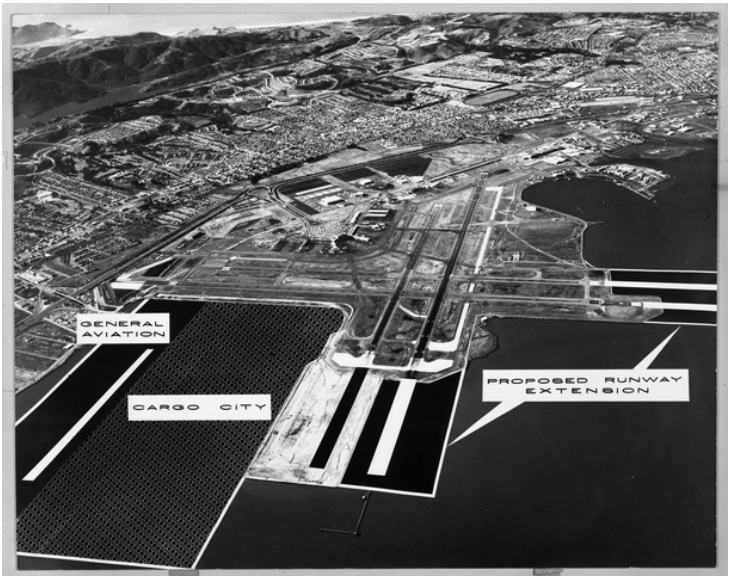
1. ***Lists and thumbnail views for route steps, galleries and public art works along a route and objects from the SFO Museum Aviation Collection*** `https://collection.sfomuseum.org` ***related to a flight arriving at or departing from SFO have been standardized and moved in to modal***

dialogs.

2. ***When supported by the browser the Web Share API*** `https://developer.mozilla.org/en-US/docs/Web/API/Web_Share_API` ***has been enabled for linking to (and sharing) individual routes.***

3. ***The new "nearby" feature*** `https://millsfield.sfomuseum.org/wayfinding/nearby/` ***allows you to find galleries, public art works and other waypoints close to arbitrary points inside the SFO terminal complex.***

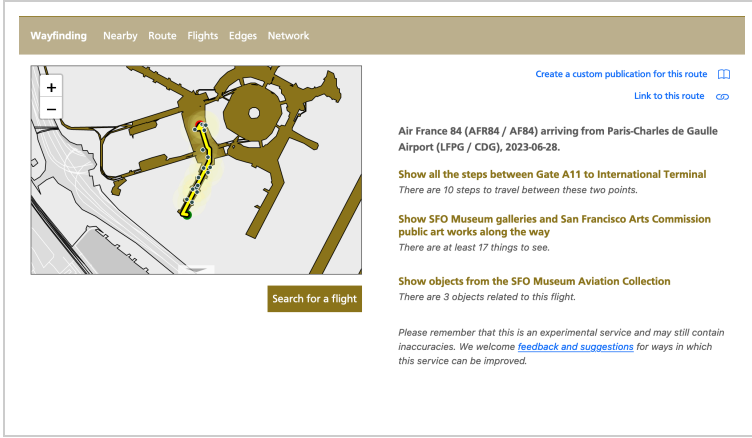
Interface changes



Negative: San Francisco International Airport (SFO), aerial with annotations. Negative. Transfer from San Francisco

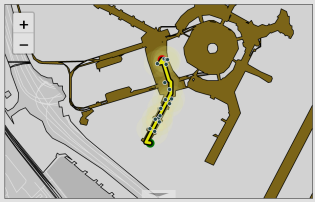
International Airport, SFO Museum Collection. ***2011.032.0957***

The most noticeable change will be updates to the wayfinding interface. Lists and thumbnail views for route steps, galleries and public art works along a route and objects from the **SFO Museum Aviation Collection** `https://collection.sfomuseum.org/` related to a flight arriving at or departing from SFO have been standardized and moved in to modal dialogs. Instead of trying to squeeze multiple lists, usually of different lengths, in to parent elements with fixed sizes and scrolling controls, a short description of each list is displayed and, when clicked, launches a modal dialog that can take full advantage of the available screen real estate.



For example, now that collection objects related to flights in and out of SFO are displayed in a stacked list, rather than three-abreast, there is ample space to display complete titles and credit lines for each object.

Wayfinding Nearby Route Flights Edges Network



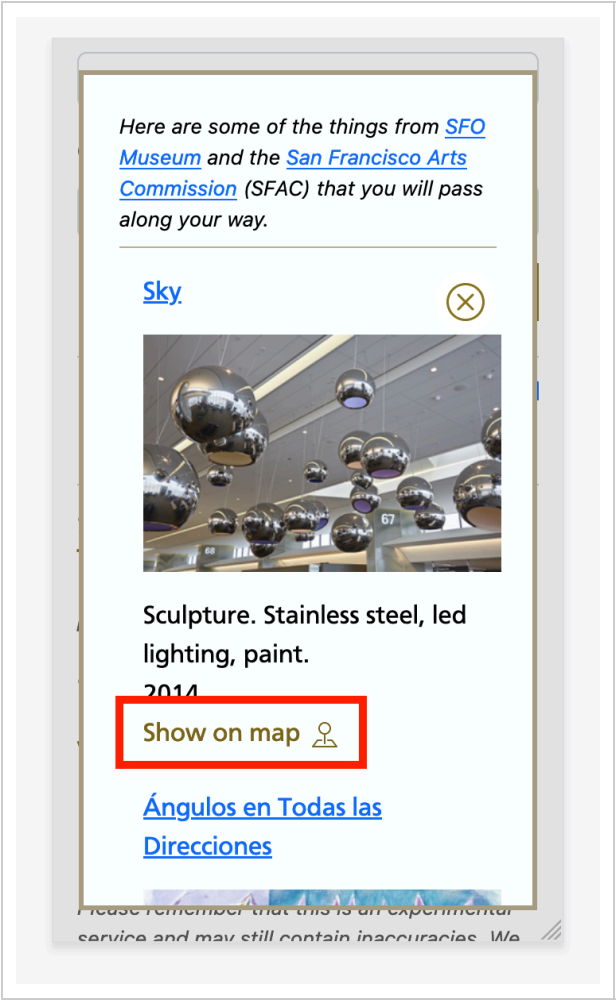
Search for a flight

Here are some objects from the SFO Museum Aviation Collection that are related to one or more aspects of this flight.

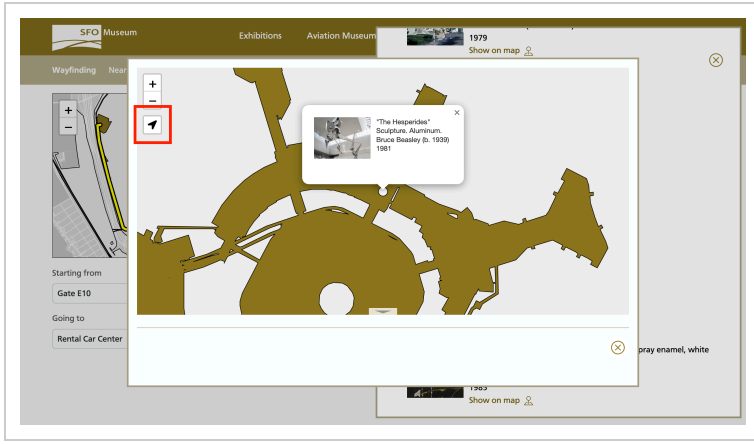
photograph: San Francisco International Airport (SFO), Central Terminal construction
1982
Transfer from San Francisco International Airport
2005.137.034.001

postcard: Air France
late 1940s-1950s
Gift of Thomas G. Dragges
2015.166.0156

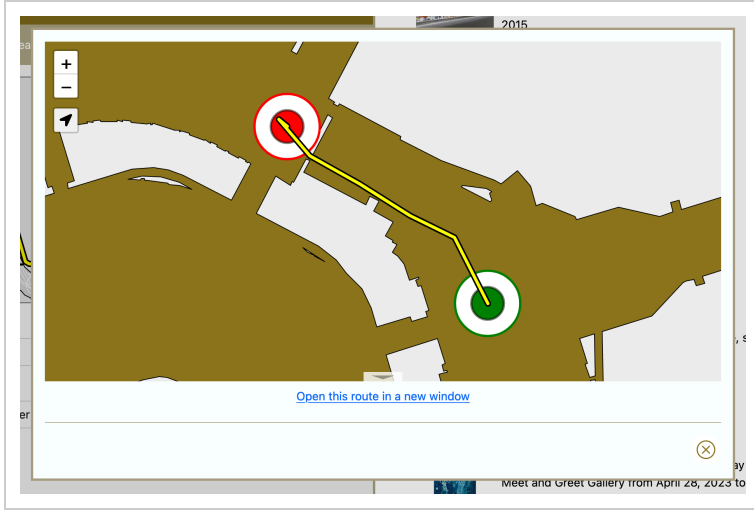
postcard: Charles de Gaulle Airport, Roissy
1970s
Gift of Thomas G. Dragges
2015.166.2185



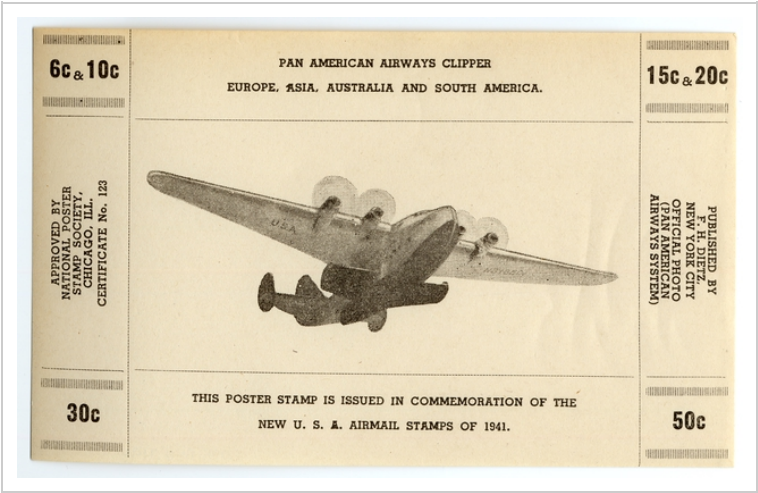
Every item in these new lists also has a new "Show on map" link which will open a second modal dialog showing just that item and its location on a map of SFO. These maps have a "geolocate" button that will, assuming you are physically located in the SFO terminal complex, use your current location to generate a route to the specific item you are looking at.



Technically, what's really happening is the site is determining the closest waypoint to your location and using that as the starting point for the new route. This is how the new "nearby" feature works. Both the nearby feature and the details of how the geolocation functionality work are discussed later in this blog post.



WebShare API



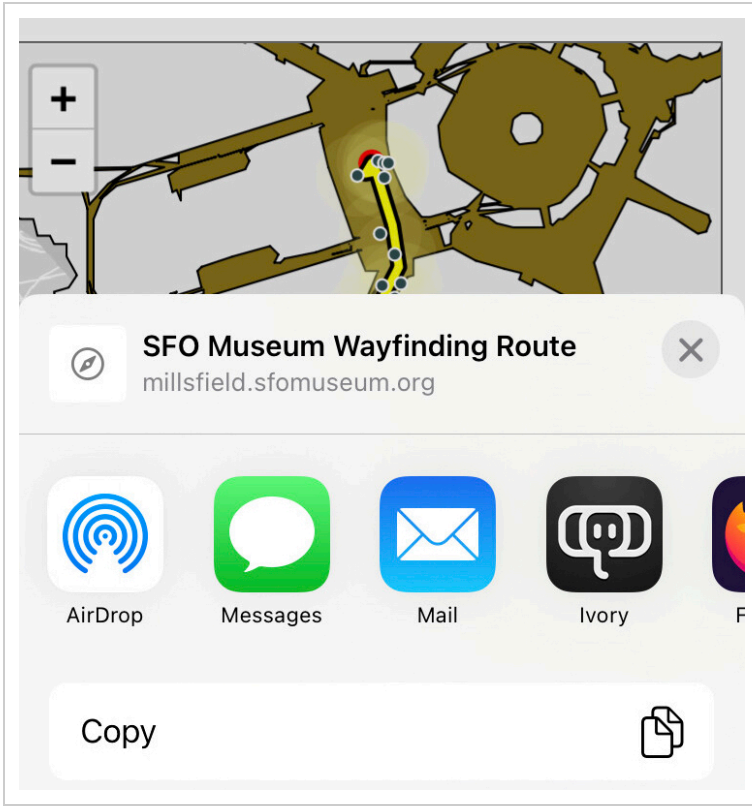
Postage stamp: Pan American Airways, Clipper poster stamp. Ink, paper. Gift of Jon E. Krupnick, SFO Museum Collection. ***2008.056.0498***

In browsers that support it we've enabled the **Web Share API** `https://developer.mozilla.org/en-US/docs/Web/API/Web_Share_API` which is described as a programmatic interface to:

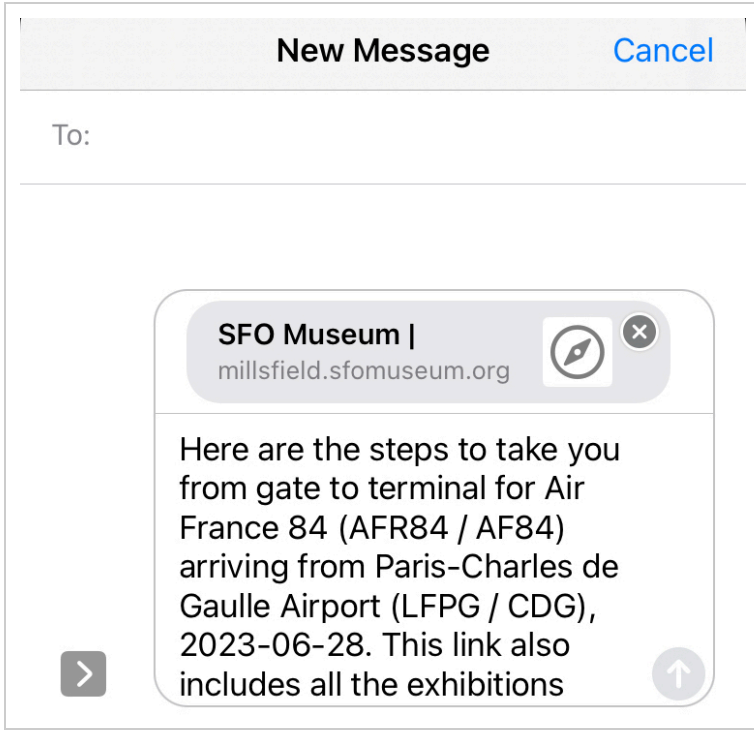
> ...allow a site to share text, links, files, and other content to user-selected share targets, utilizing the sharing mechanisms of the underlying operating system. These share targets typically include the system clipboard, email, contacts or messaging applications, and Bluetooth or Wi-Fi channels.

If supported a new `"Share this route"` link will be displayed over the default `"Link to this route"` link:

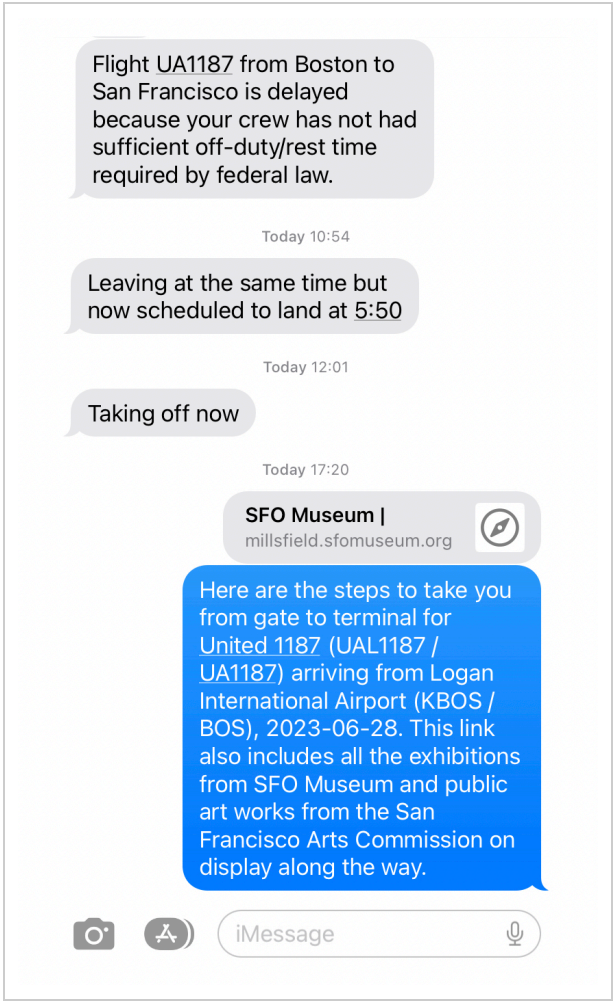
When you click the link it will display a menu with all the different ways your device supports for "sharing" things like text and images and links. For example:



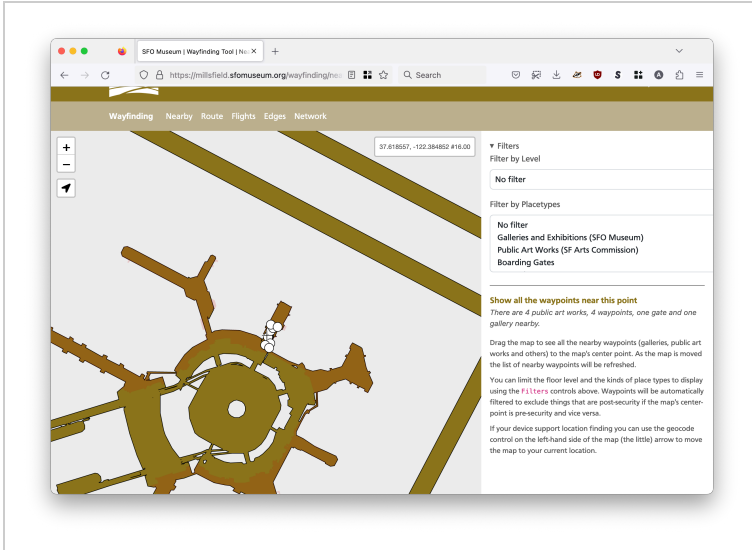
The link itself will also have a long-form description of the route being shared. For example here's the text that's included when sharing the route, through the terminals at SFO, for Air France flight 84 arriving in San Francisco as a text message:



Here's a screenshot of a conversation that happened while I was writing this blog post:

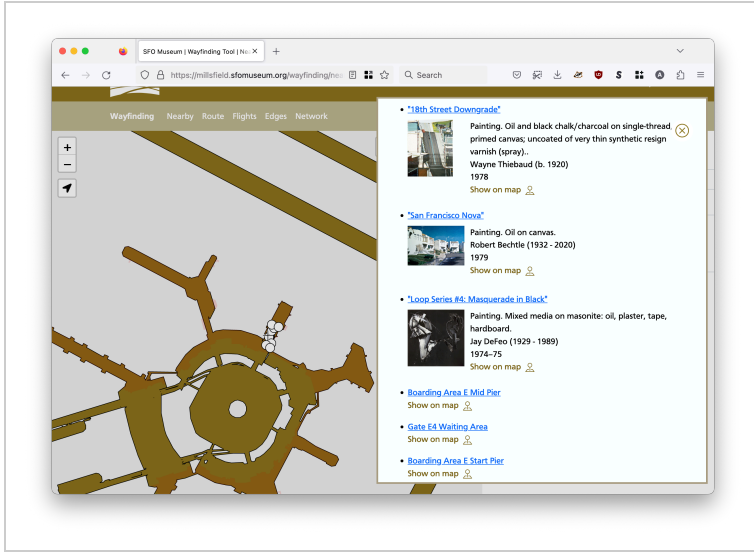

Flight UA1187 from Boston to San Francisco is delayed because your crew has not had sufficient off-duty/rest time required by federal law.
Today 10:54
Leaving at the same time but now scheduled to land at 5:50
Today 12:01
Taking off now
Today 17:20
SFO Museum |
millsfield.sfomuseum.org
Here are the steps to take you from gate to terminal for United 1187 (UAL1187 / UA1187) arriving from Logan International Airport (KBOS / BOS), 2023-06-28. This link also includes all the exhibitions from SFO Museum and public art works from the San Francisco Arts Commission on display along the way.
iMessage

Nearby



The new "nearby" feature allows you to define an arbitrary point inside the SFO terminal complex and list all the SFO Museum galleries and exhibitions, public art works from the San Francisco Arts Commission and other waypoints near that location. You can limit results to a specific floor level or by their place type and waypoints will be automatically filtered to exclude things that are post-security if the map's center-point is pre-security and vice versa. You can try the "nearby" feature here:

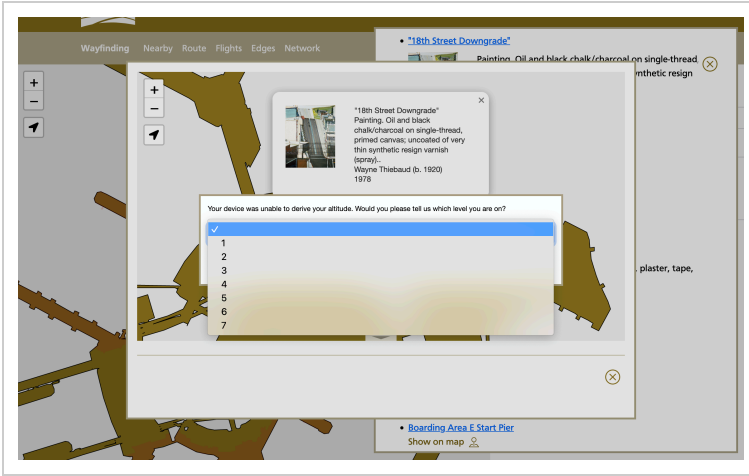
https://millsfield.sfomuseum.org/wayfinding/nearby/ `https://millsfield.sfomuseum.org/wayfinding/nearby/`



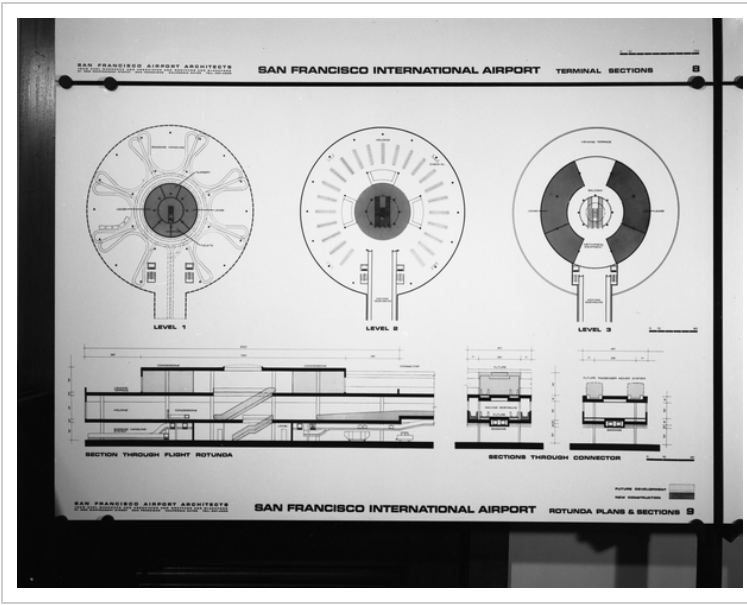
You can also link to a specific spot using the "nearby" service by appending the relevant latitude, longitude and map zoom level to the URL. For example here are all the things near the Gate E6 waiting area in Terminal 3:

https://millsfield.sfomuseum.org/wayfinding/nearby/#16/37.6187/-122.3848 `https://millsfield.sfomuseum.org/wayfinding/nearby/#16/37.6187/-122.3848`

One of the challenges of designing a "nearby" system at a place like SFO is that the terminal complex spans multiple levels each with its own set of waypoints. A web browser's **geolocation functionality** `https://developer.mozilla.org/en-US/docs/Web/API/Geolocation` will, when available, returns an altitude property indicating how many meters above sea level it thinks the device doing the geolocating is. With that information we can make a good faith estimate about which level of the terminal complex a nearby query is being made from.



It is often the case, however, that elevation data is *not* included with geolocation events. When that happens the wayfinding site will interrupt the geolocation flow and ask you to specify what level of the SFO terminal complex you are on. I mentioned that when a browser *does* return altitude data we make a "good faith estimate" to associate that elevation with the relevant level at SFO. As of this writing there may still be errors in our approach because elevation data do not apply equally across all parts of the airport. That is the floors and ceilings for "Level 2" in the International Terminal have different heights than the "Level 2" in Terminal 1. We are waiting on more accurate numbers from the airport itself but it points to the need to develop an elevation-aware **point-in-polygon** `https://millsfield.sfomuseum.org/blog/tags/pointinpolygon` interface (an interesting technical challenge, albeit one I kind of wish we could avoid) and corresponding data attributes to account for these differences.



Negative: San Francisco International Airport (SFO), architectural rendering. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.1947***

It is also worth noting that consumer grade elevation data is not very precise either as demonstrated by the tool we developed to collect the data in use today (discussed below) so in the end it may be easiest to *always* interrupt the geolocation flow to prompt visitors for the level they want to scope queries to. We'll see how things go but we welcome **any**
feedback `https://millsfield.sfomuseum.org/wayfinding/about/#feedback` you have using this new feature.

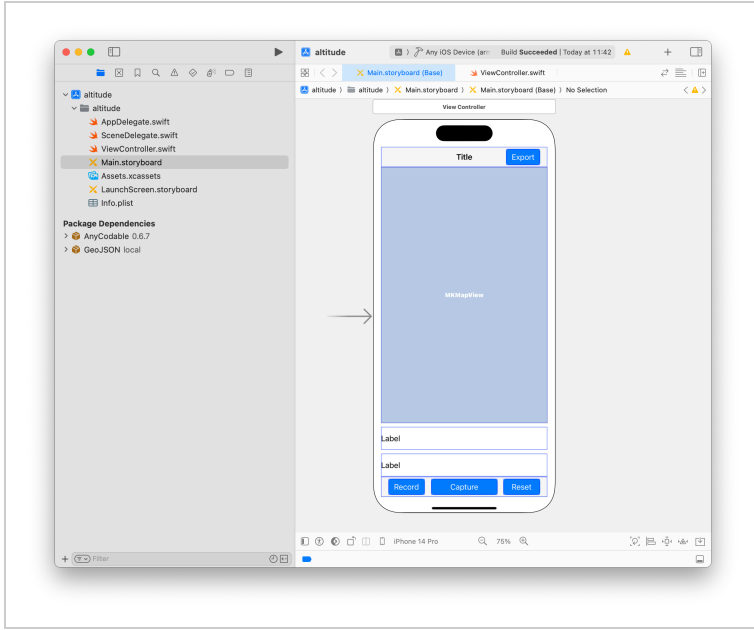
Under the hood

Negative: San Francisco International Airport (SFO), road construction. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.2480***

The rest of this blog post gets in to some of the technical details of how we implemented these updates. If that's not of interest you can stop reading now and we invite you to investigate the interface changes we've made and try the new "nearby" feature here:

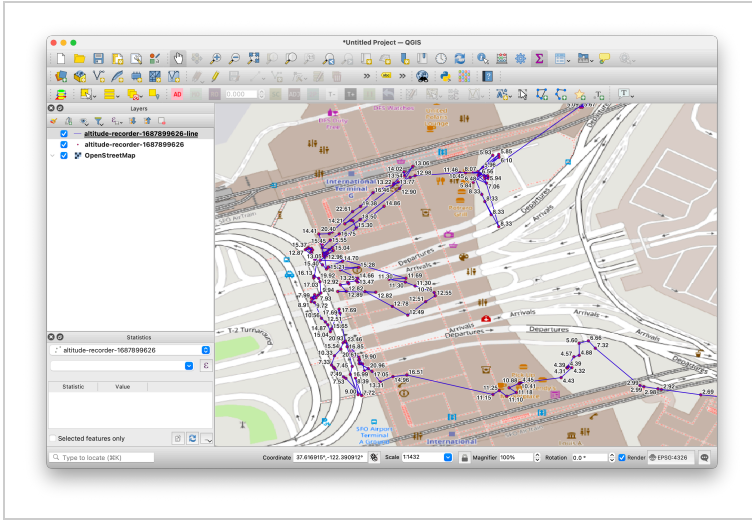
https://millsfield.sfomuseum.org/wayfinding/nearby/ `https://millsfield.sfomuseum.org/wayfinding/nearby/`

ios-altitude-recorder



As mentioned, SFO Museum needed a tool to measure, or more precisely to approximate, the different elevations in the SFO terminal complex in advance of more accurate data being delivered by the airport itself. `ios-altitude-recorder` is that tool. It is an iOS data collection application designed to record location information as it is generated by the operating system's **CoreLocation** `https://developer.apple.com/documentation/corelocation` framework. Locations can be recorded explicitly in atomic units or automatically as they are received. Data is persisted using the **SwiftData** `https://developer.apple.com/documentation/swiftdata` framework (which means the application only works with iOS 17 or later) and it is possible to export location records as a **GeoJSON** **FeatureCollection** `https://tools.ietf.org/html/rfc7946` .

That's all the application does. It is being released as-is in the spirit of generousity in the hopes that it might be useful to other museums and cultural heritage institutions who need to collect similar data or simple as a reference implementation for how to develop bespoke location-based iOS applications.



Please remember that **indoor location continues to be imprecise** `https://millsfield.sfomuseum.org/blog/2018/08/07/old-maps/` , as you can see from the screenshot above where I often appear to be floating in mid-air or standing in the middle of the road when I was actually standing indoors, and you should factor that knowledge in to how you use and interpret the data produced by `ios-altitude-recorder`. The source code for the application is available from **SFO Museum's GitHub account** `https://github.com/sfomuseum/` :

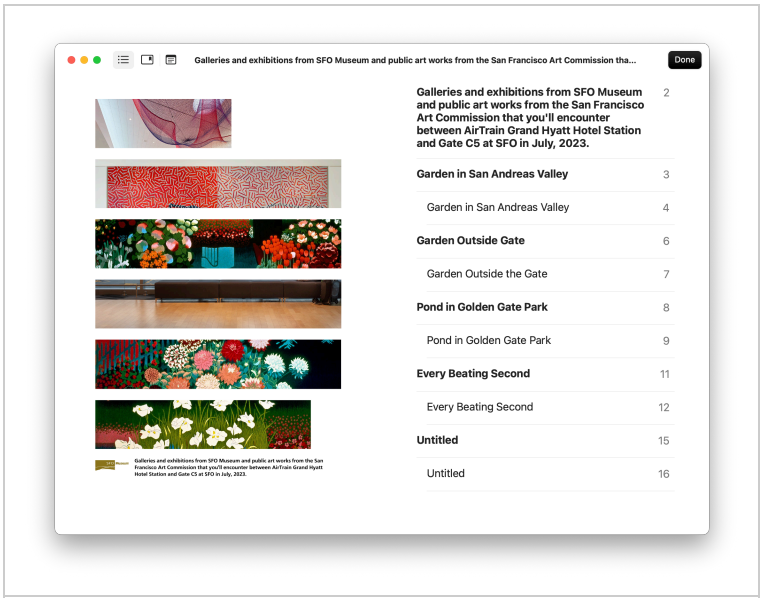
https://github.com/sfomuseum/ios-altitude-recorder `https://github.com/sfomuseum/ios-altitude-recorder`

2023-07-03

things I have written about elsewhere #20230710

Updating the SFO Museum Wayfinding Service -...

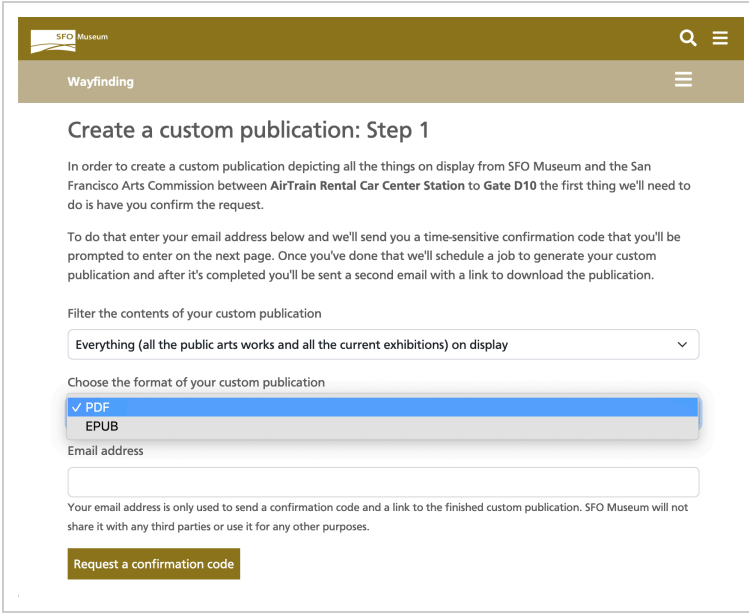
Updating the SFO Museum Wayfinding Service - EPUB (ebook) publications



This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/07/10/wayfinding-epub` , in July 2023.

This is a short blog post to announce that we have enabled support for generating publications of all the SFO Museum galleries and exhibitions and public art works from the San Francisco Arts Commission along a **wayfinding route** `https://millsfield.sfomuseum.org/wayfinding` as EPUB documents.

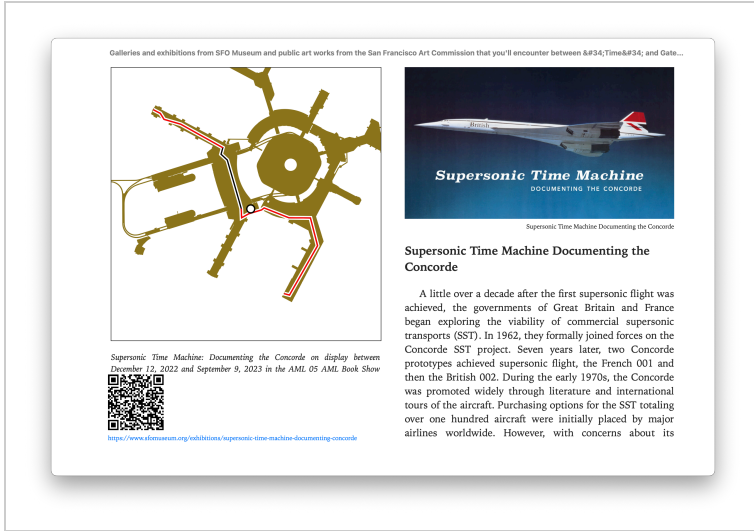
EPUB `https://en.wikipedia.org/wiki/EPUB` is a technical standard for publishing e-books that has been adopted by most electronic book devices on the market today.



The introduction of the EPUB format changes two things in the **SFO Museum Wayfinding application** `https://millsfield.sfomuseum.org/wayfinding` :

1. *There's now a menu option labeled* `EPUB` *when you request a new publication.*

2. *You'll receive an EPUB-formatted publication when you select it.*



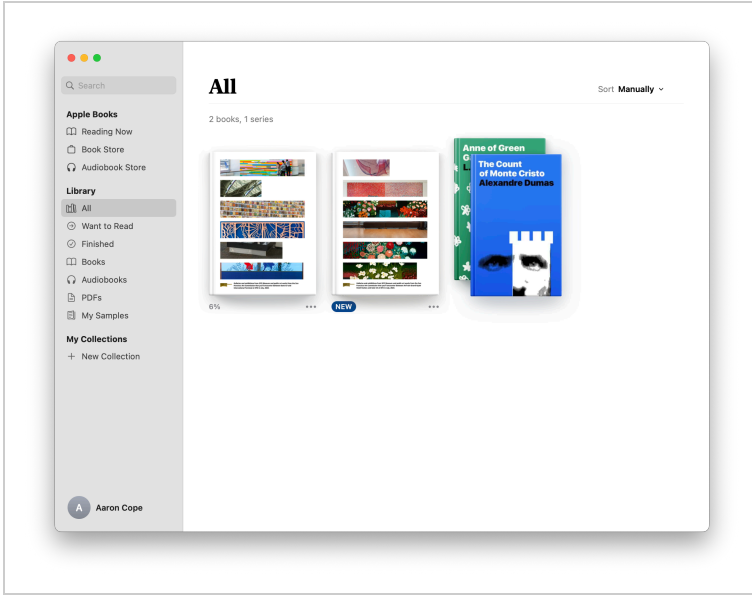
That's it. EPUB publications have all the same images and maps and texts that are included in PDF publications and each one has a custom cover derived from the objects and exhibitions that you'd encounter along a wayfinding route. Most of the work to support EPUB publications was done behind the scenes to adapt the code we already used to produce one publication format (PDF files) to support two formats and, if that work was successful, make it easier and faster to support others in the future.

In the **Updating the SFO Museum Wayfinding Service - Publications** `https://millsfield.sfomuseum.org/blog/2023/06/12/wayfinding-pdf/` blog post I wrote:

The initial launch of publications only supports one format: PDF files. In the coming weeks we plan to add support for another:

EPUB https://en.wikipedia.org/wiki/EPUB , a commonly used format for e-books. PDF files have broad support in all mobile and desktop computers and are familiar to most people. It's also well-suited to printing since that is the heritage the format springs from. Some people have criticized PDF files because they are a less open format than others (like EPUB) and because they typically are designed for letter-sized pieces of paper that may not display well on mobile devices. This is one of the situations where everyone is right but we had to start somewhere and we chose to start with PDF files. The goal for these publications is to support as many different formats as people desire (and is practical to produce).

So this is our first step to address that challenge. **We'd love to hear what other formats you think we should support** https://millsfield.sfomuseum.org/wayfinding/about/#feedback .

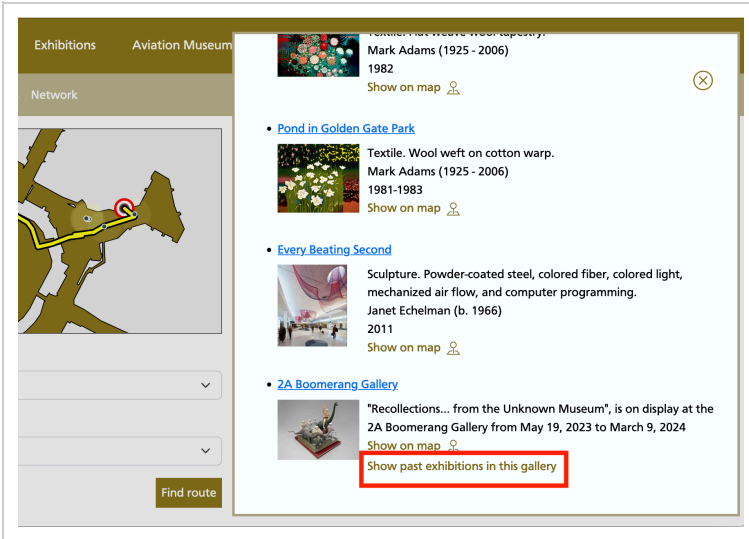

Search
Apple Books
Reading Now
Book Store
Audiobook Store
Library
All
Want to Read
Finished
Books
Audiobooks
PDFs
My Samples
My Collections
New Collection
A
Aaron Cope
All
2 books, 1 series
Sort Manually
6%
NEW
Anne of Green
The Count
of Monte Cristo
Alexandre Dumas

2023-07-10

things I have written about elsewhere #20230816

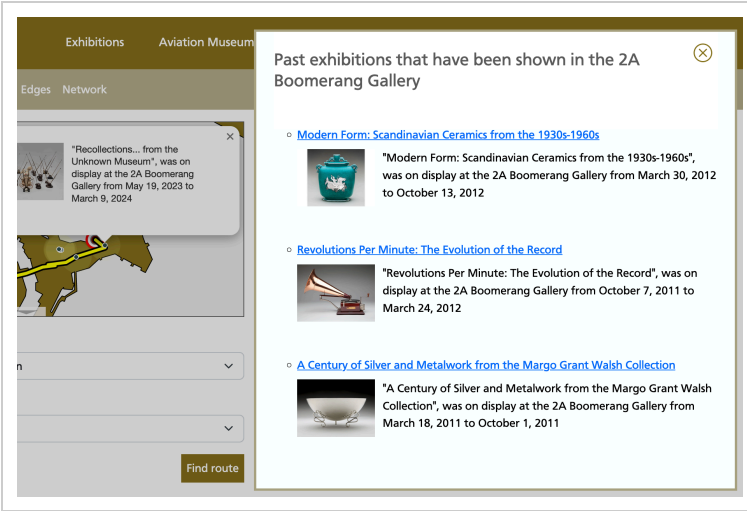
Updating the SFO Museum Wayfinding Service -...

Updating the SFO Museum Wayfinding Service - Past exhibitions for galleries along a route

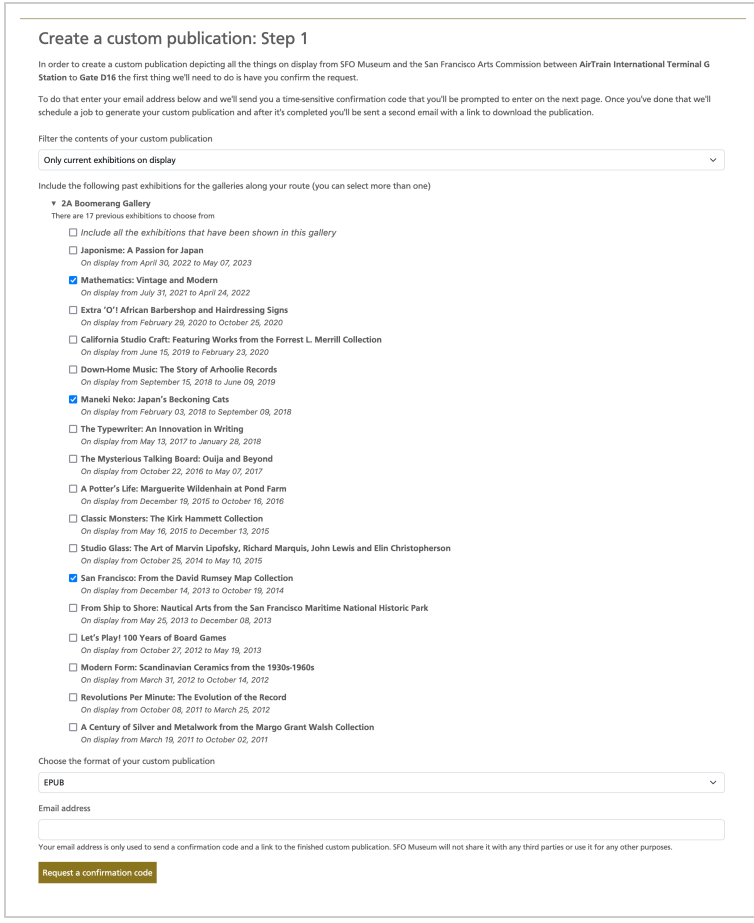


This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/08/16/wayfinding-past/` , in August 2023.

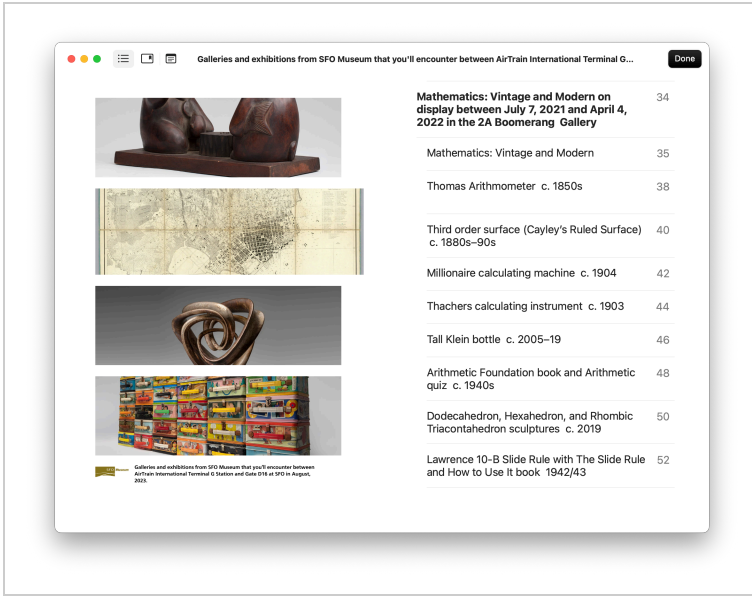
This is a short blog post to tell you that the **SFO Museum Wayfinding Service** `https://millfield.sfomuseum.org/wayfinding` has been updated to display past exhibitions for galleries along a wayfinding route. Additionally, one or more past exhibitions for a gallery can be included in the custom PDF and EPUB publications for a route, as described in **earlier blog posts** `https://millsfield.sfomuseum.org/blog/tags/wayfinding` .



The modal list views for galleries and exhibitions along a route has been updated, where appropriate, to include a `See past exhibitions` link. Clicking that link will open a second modal dialog listing all of the past exhibitions that have been shown in that gallery. Each exhibition links to their respective page on the **sfomuseum.org** `https://www.sfomuseum.org/exhibitions/past` website. If you choose to create a publication for a wayfinding route the controls for doing so now include a list of all the galleries with past exhibitions. Each gallery is a collapsible menu cataloging all of the exhibitions shown in that space. You can choose to include just a selection of exhibitions or, if you want, all of them!

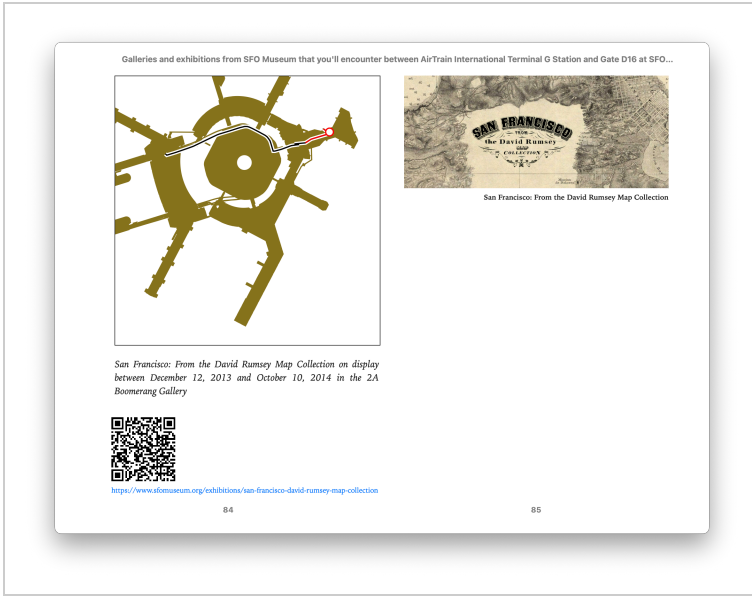


Selecting all the past exhibitions for all the galleries along a route might yield a very long (and large) publication but that's sort of the point. Nothing would make us happier than to have people create a custom publication, hundreds and hundreds of pages in length, of exhibitions that SFO Museum has produced over the years to take with them and read at home or on a long flight.



To illustrate that idea, here's **a link to the (EPUB) publication shown in the screenshots** `images/sfomuseum_wayfinding_publication_20230811_4VxsKHmA8j1.epub` (20MB) that accompanies this blog post. This publication includes all the **current exhibitions** `https://www.sfomuseum.org/exhibitions/unknown-museum` on display between the **G-side AirTrain Station in the International Terminal and Gate D16** `https://millsfield.sfomuseum.org/wayfinding/#from=1796982275&to=1763588157` as well as the **Mathematics: Vintage and Modern** `https://www.sfomuseum.org/exhibitions/mathematics-vintage-and-modern` , **Maneki Neko: Japan's Beckoning Cats** `https://www.sfomuseum.org/exhibitions/maneki-neko-japans-beckoning-cat` and the **San Francisco: From the David**

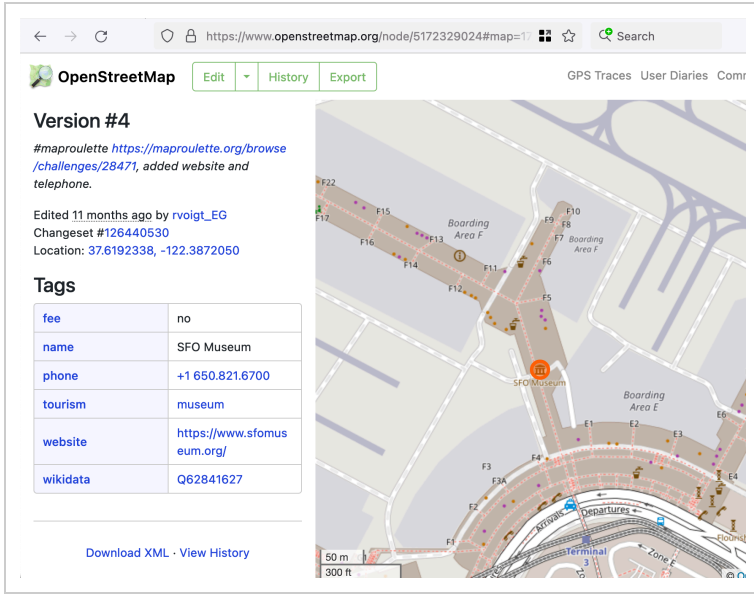
Rumsey Map Collection `https://www.sfomuseum.org/exhibitions/san-francisco-david-rumsey-map-collection` exhibitions that were shown in the "Boomerang" Gallery in Terminal 2.



SFO Museum has been producing exhibitions since 1980 and if people sometimes have trouble wrapping their heads around what's on display *today* it's easy to imagine that most people have no idea what we've done in the past (and that we've been doing it for over 40 years). By including past exhibitions in wayfinding routes and publications we're hoping to show people not only all the cool and interesting stuff we've displayed over the years but also the breadth and velocity of that work to help reinforce the idea that "the airport *is* the museum".

*Installation view of the **The Art of Recology: The Artist in Residence Program 1990-2013** exhibition, on display in the 3C North Connector Gallery between March and October 2013.*

As part of that work we hope to update the wayfinding service, shortly, to allow for the inclusion of past galleries no longer on display in the terminals at SFO. If you've ever traveled through SFO at any time since 1981 there's a good chance you've walked the long hallway in Boarding Area F between the security checkpoint and the gates and seen at least one of the almost 100 exhibitions that have been shown in the "3C North Connector" gallery.



*The "3C North Connector" gallery classified as "the museum" in **OpenStreetMap** (OSM) in August 2023, despite having been closed in 2020. That's not a slight against OSM but is a good illustration of how passengers understand and perceive how SFO Museum is situated at the airport.*

The gallery closed in 2020 but that long hallway remains and is traveled by thousands of people a day and we'd like to give them a way to see all the past exhibitions they are walking "through". Even if the entire airport is actually the museum the reality is that for many people that one gallery was, and remains, the museum in their minds so it seems only right that we include it in the wayfinding service.

Cocktail napkin: Southwest Airlines. Paper, ink. Anonymous gift, SFO Museum Collection. ***2018.135.006***

In the meantime, if you are in the San Francisco Bay Area I will be speaking more about the SFO Museum Wayfinding project, along with a number of other speakers, at the **California Map Society Conference** `https://californiamapsociety.org/event-5296754` on August 19, at Stanford University. If you are not in the Bay Area you can also register to attend the conference **virtually over Zoom** `https://www.eventbrite.com/e/california-map-society-conference-tickets-634245534467` .

Wayfinding at SFO Museum
California Map Society / August 2023
Aaron Straup Cope, SFO Museum

2023-08-16

saying the quiet part out loud

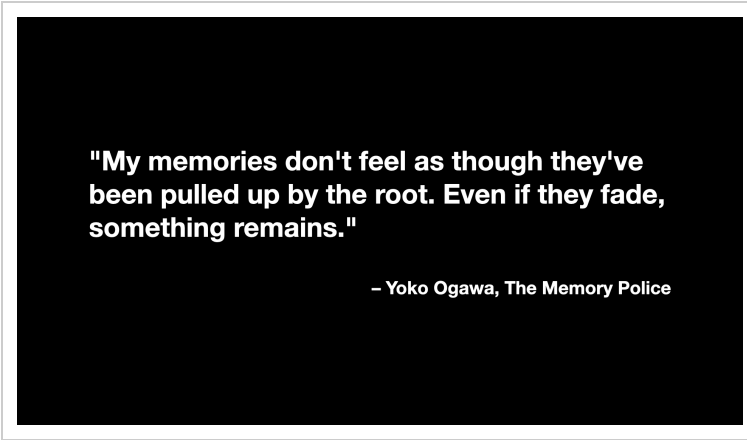
Wayfinding at SFO Museum

Wayfinding at SFO Museum

Photograph: San Francisco Airport, model *1936*
Transfer from San Francisco International Airport
Collection of SFO Museum ***1997.52.064.017***

Last weekend I had the opportunity to speak about ***SFO Museum's Wayfinding system*** `https://millsfield.sfomuseum.org/blog/tags/wayfinding` *at the* ***2023 California Map Society Conference*** `https://californiamapsociety.org/event-5296754` *, held at the* ***David Rumsey Map Center*** `https://library.stanford.edu/libraries/david-rumsey-map-center` *. Thanks to* ***Tom Paper*** `https://www.aag.org/profile/tom-paper/` *, who organized the event, for inviting me and allowing me to talk about maps but really to talk about what these particular maps makes possible and what those possibilities say about the museum sector in 2023.*

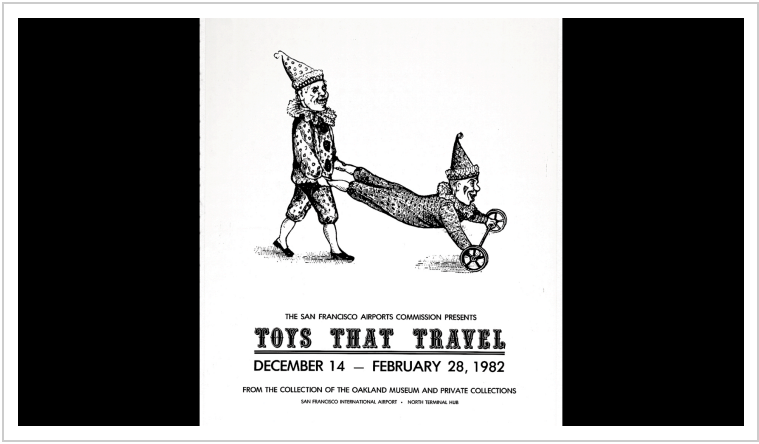
This is what I said.



I'd like to start with a quote from Yoko Ogawa's **"The Memory Police"** `https://en.wikipedia.org/wiki/The_Memory_Police` :

> "My memories don't feel as though they've been pulled up by the root. Even if they fade, something remains."

This is a book that wouldn't suffer from being required reading for all Museum Studies students.



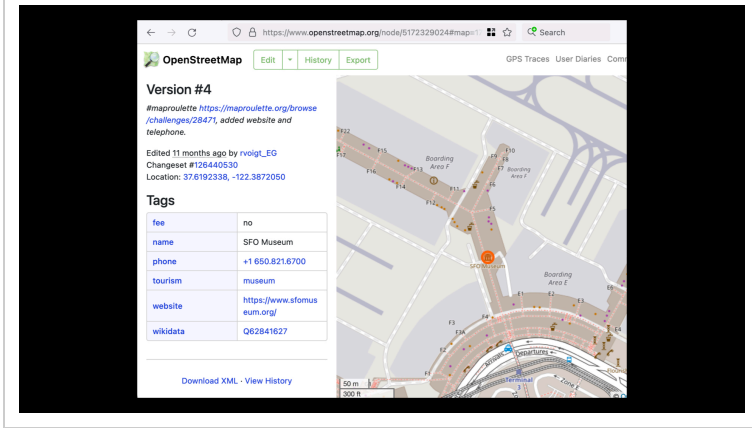
Exhibition poster: San Francisco Airports Commission, Toys That Travel *1981*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2017.079.011***

Before the COVID-19 pandemic the **San Francisco International Airport** `https://flysfo.com/` , or SFO, handled over 58 million passengers a year and, if the current trends continue, the airport is well on its way to matching and exceeding those numbers in the years to come.

In 1980 the airport started an arts program that would become **a fully accredited museum** `https://sfomuseum.org/` in 1999. In those 43 years the museum has mounted over 1,500 exhibitions in dozens of galleries, stewards over 100 public artworks purchased by the San Francisco Arts Commission and has built **a permanent collection of over 150,000 aviation-related objects** `https://collection.sfomuseum.org/` .

For these reasons I am fond of pointing out that the **San Francisco**

International Airport Museum `https://sfomuseum.org/` is the busiest museum in the world... that no one has ever heard of.



This screenshot was taken from the OpenStreetMap website `https://www.openstreetmap.org/node/5172329024` a couple of days ago. It depicts the "3C North Connector" gallery that was opened in 1981 but was removed, three years ago, in 2020. For many people this one gallery is the museum. If it's not this gallery it's usually another specific gallery located in whatever terminal they travel out of most often. The Museum publishes online and printed maps detailing the scale and scope of its holdings but, despite all that, the question those of us who work at the museum are still most often asked is : *Where is the museum?*

We try to tell people that the whole airport is, in fact, the museum but for people not used to the idea of a museum in an airport, in the first place, this kind of expansive answer can be even more confusing.

Negative: San Francisco Airport, passenger waiting room *1954*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2011.032.0406***

Did you know that in 2015 you could travel **from one end of Boarding Area G to the other end of Boarding Area E** `https://millsfield.sfomuseum.org/wayfinding/#from=1763588467&to=1763588357` , post-security? Or that in 2023 you can go all the way to **the end of Boarding Area C now** `https://millsfield.sfomuseum.org/wayfinding/#from=1763588467&to=1763588255` ? Or that in 2024 you'll be able to do the full loop of all the terminals post-security?

Did you know that if your flight is departing from Boarding Area G and it is delayed by an hour you have time to walk and see **a painting by Wayne Thiebaud** `https://www.sfomuseum.org/public-art/public-collection/18th-street-downgrade` , **a mural by Paul Madonna** `https://www.sfomuseum.org/public-art/public-collection/here-you-are-are-you-here` , works on loan from

The Unknown Museum in Mill Valley https://www.sfomuseum.org/exhibitions/unknown-museum and **a sculpture by Yayoi Kusama** https://www.sfomuseum.org/public-art/public-collection/high-heels-going-heaven ?

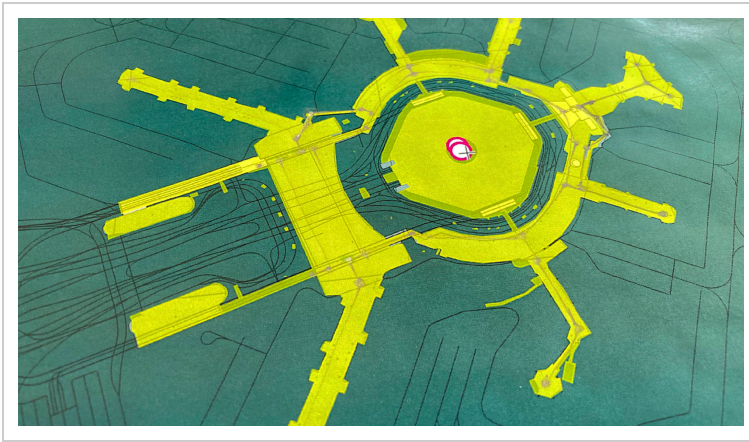
It's okay if you didn't. Most people don't. This is both the challenge and the opportunity of the SFO Museum. I say it's a challenge because at a very practical level we are competing for people's time and attention. People who are racing to catch a flight, struggling with luggage, trying to keep small children occupied or all the above all surrounded by other people who are all doing the same thing.

Put bluntly people are so busy dealing with the mechanics of travel that they don't even know that they did not notice the Museum even though we are all around them.

The opportunity, for SFO Museum, is that a non-trivial number of those 58 million passengers are going to come back to SFO, at least once. Our opportunity rests in the fact that we are able to say to people: It's okay if you missed something on your first pass through the terminals. It will still be here when you return.

Name pin: Virgin America, Art *2007*
Gift of Art Huggard
Collection of SFO Museum ***2019.066.014***

This is a talk about maps because a map is the device we've chosen to use to try and address the problems I've just described. But it's really a talk about what the map makes possible and what those possibilities say about the museum sector in 2023. I am going to show you the work we've done but I am going to show you this work quickly and probably only skim the surface in order to discuss those larger issues, and why they are important, at the end of this talk.



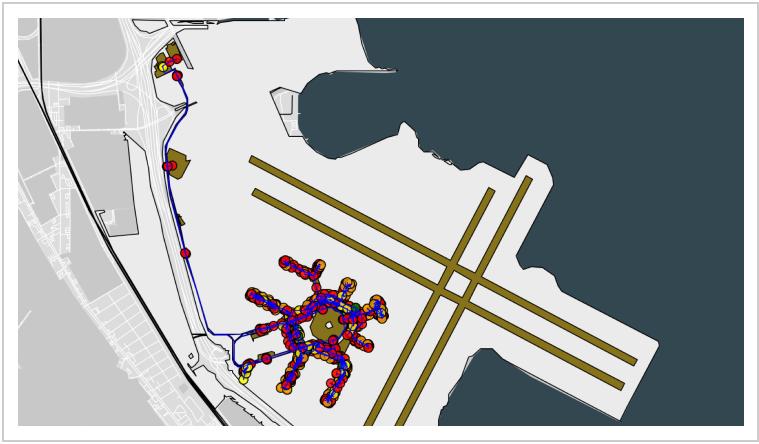
If the goal is to promote discovery of the Museum's programming in the terminals the first questions we need to answer are: Where is a passenger? Where is the thing they are trying get to? How do they travel between those two points?

In a word: Wayfinding, or routing. Which sounds pretty straightforward in principle until you look at the burden and overhead of running routing software, little of which was developed for a complex space like an airport.

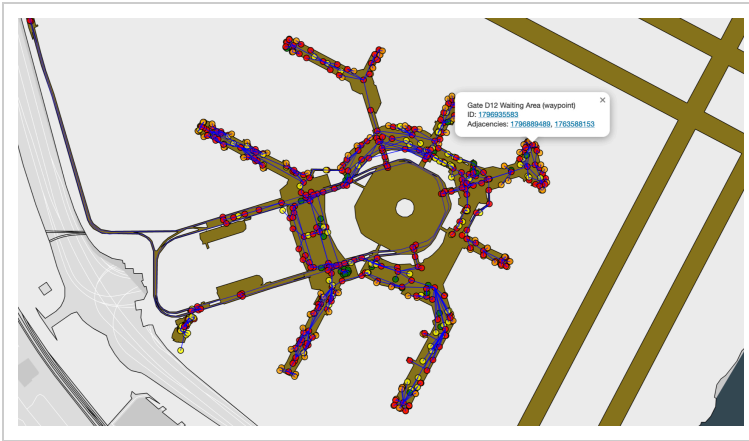
It's very easy to get overwhelmed and discouraged by the scale of the problem. But what if it didn't need to be that complicated? More specifically, what if we didn't need a wayfinding system precise enough to be used by automated systems? What if instead we married a very simple wayfinding system with just enough detail to direct a passenger to the general area where a gallery or object is located and then let that same passenger figure it out from there?

This is an early sketch of what I imagined that wayfinding system

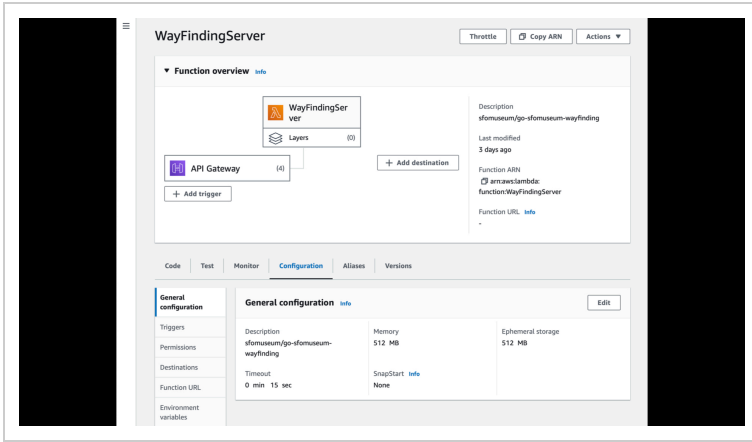
might look like with a network of about two dozen connected waypoints.



In reality it looks like this. As of today there are **about 250 waypoints** `https://github.com/sfomuseum-data/sfomuseum-data-wayfinding` and another **150 Museum-specific venues and spaces** `https://github.com/sfomuseum-data/sfomuseum-data-architecture/` you might be routed to. That's not too bad for something the size of the SFO terminal complex.

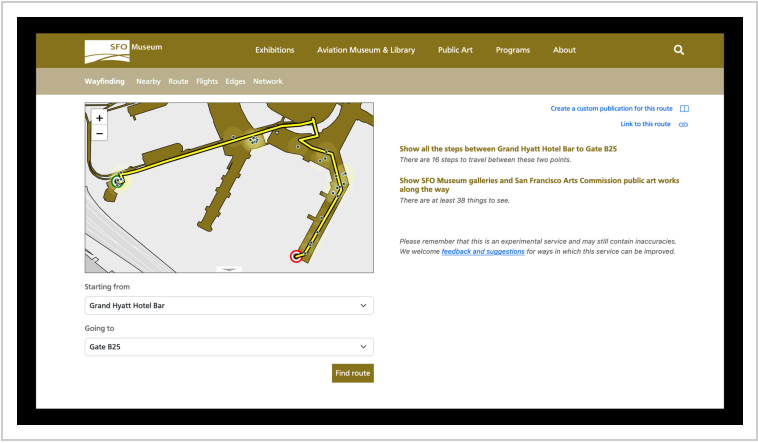


Red dots are waypoints. Orange dots are gates. Green dots are galleries and yellow dots are public art works. It's hard to see in this image but the network also spans multiple levels at the airport.

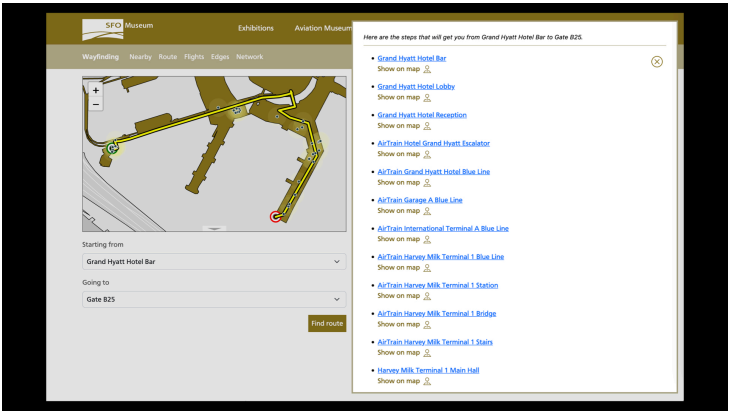


All of that data is bundled in to an Amazon Web Services Lambda function and the actual wayfinding functionality is performed using **a directed graph derived from those data, in-memory, at runtime** `https://github.com/dominikbraun/graph` . That's maybe more detail than you wanted but I mention it to point out that this service costs us pennies to run and doesn't require setting up, baby-sitting or paying for a spatial database.

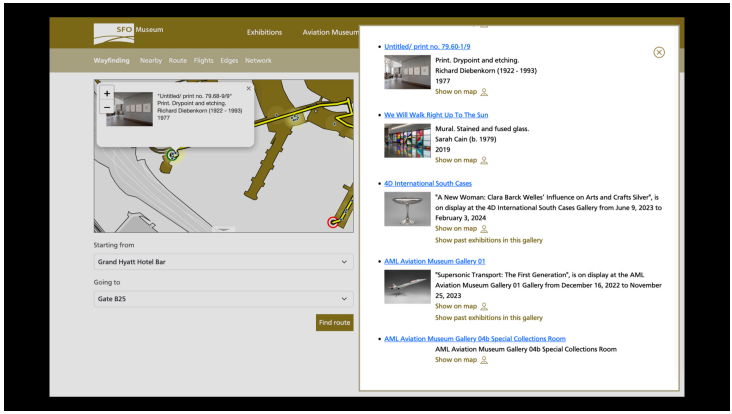
There is obviously a tipping-point in the size of a network where a spatial database becomes necessary but it's important to remember that, in 2023, the distance before which many projects reach that tipping point continues to grow.



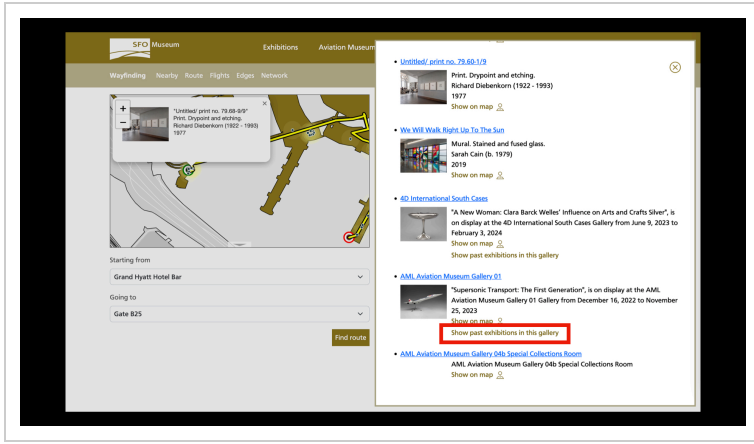
Here's what the current iteration of **the wayfinding service** `https://millsfield.sfomuseum.org/wayfinding/` looks like. Given any two waypoints we can derive the route between them as well as any Museum related content within 50 meters of that route. For example, **between the bar at the Grand Hyatt Hotel and Gate B25** `https://millsfield.sfomuseum.org/wayfinding/#from=1847571929&to=1763588201` there are 16 waypoints and 38 galleries and public art works on display.



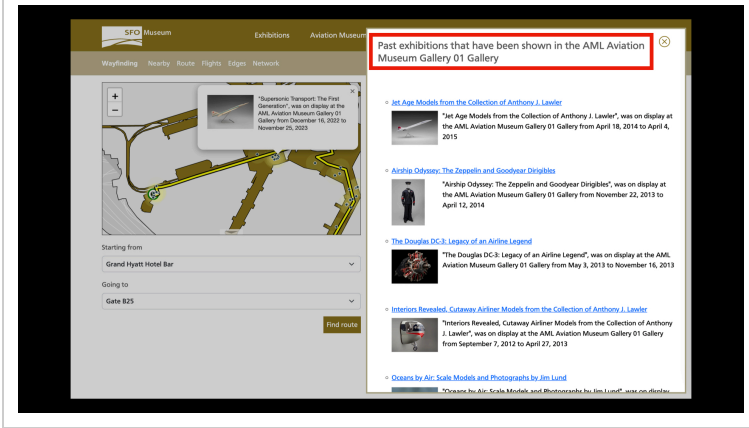
After a bit of experimentation we've settled on **using modal dialogs for displaying the details** `https://millsfield.sfomuseum.org/blog/2023/07/03/wayfinding-nearby/` for waypoints and museum programming because these translate best to mobile devices and mimic the "sheets" metaphor used in native mobile applications.



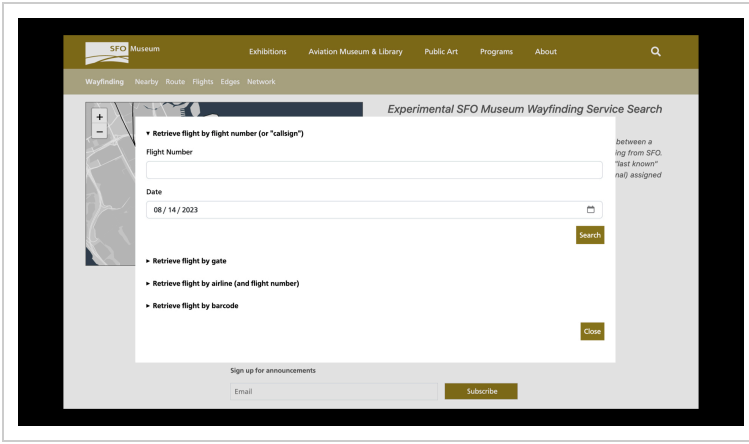
Here's a screenshot of the modal dialog used to display Museum-related programming. Did you know that there are **Richard Diebenkorn prints hanging in the lobby of the airport hotel** `https://www.sfomuseum.org/public-art/public-collection/2-3-5-7-nine-drypoints-and-etchings` ? Now you do.



Nearly every gallery in the terminals has been used to show more than one exhibition so we make sure to try and **draw people's attention to them** `https://millsfield.sfomuseum.org/blog/2023/08/16/wayfinding-past/` .

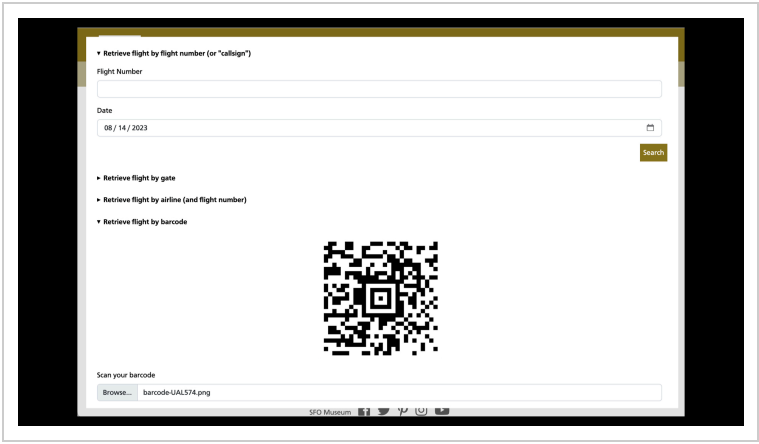


As I mentioned earlier the whole notion of a museum in a airport is still pretty foreign to most people so it's important to show them the depth and the breadth of our work. Very soon we plan to include past galleries, like the one I mentioned at the beginning of this talk, in the wayfinding maps.

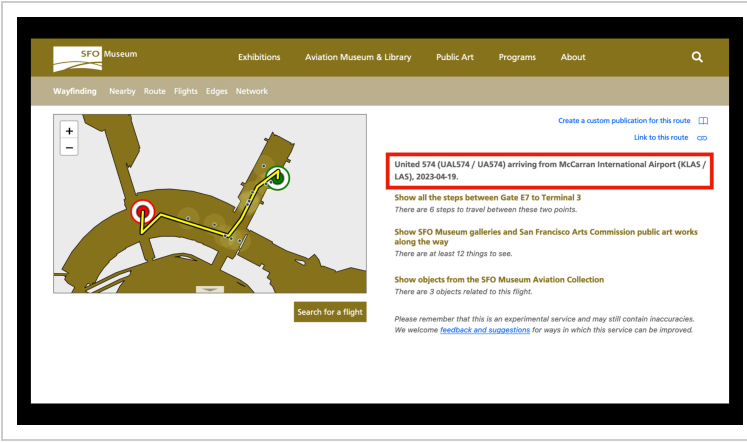


Sometimes people don't know where they are in the terminals beyond whatever is written on their boarding pass so we've also made it possible to **derive a route from a flight number** `https://millsfield.sfomuseum.org/blog/2023/05/08/wayfinding/` . We know both the gate and the terminal associated with a flight and its operating airline, respectively, so it's possible to create a wayfinding route through the airport even if a person has never been to SFO before and doesn't know their way around.

A passenger can enter their flight number manually, look it up by gate number or even by scanning the barcode on their boarding pass.

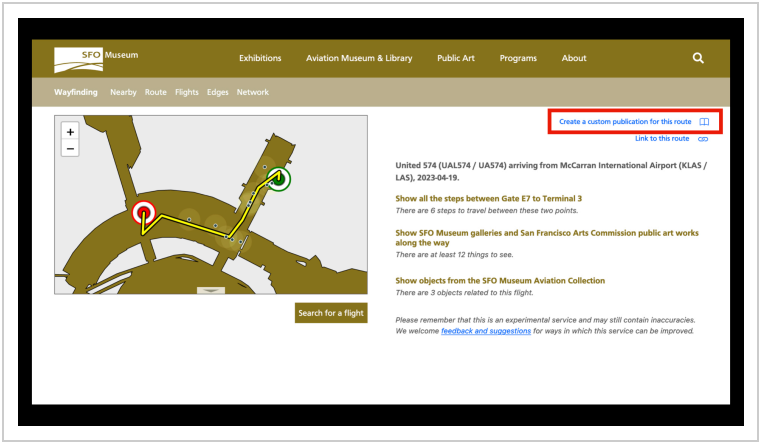


Importantly, **the barcode scanner operates entirely on-device** `https://github.com/sfomuseum/rs-sfomuseum-bcbp` and no boarding pass data is sent over the wire to the Museum or the airport or anyone else.

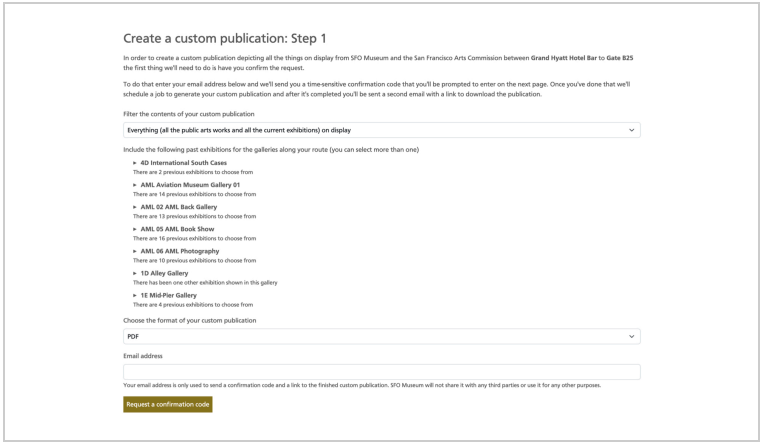


Note that the date for this route is April, 2023. We want to make sure that a person can create a wayfinding route even after they've left the airport or even months after their trip.

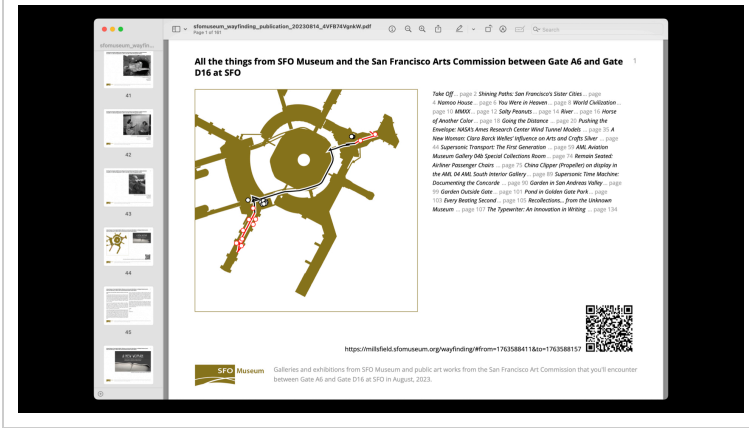
As I said earlier, we are operating in an environment where people are busy and distracted and we will not reach them unless we design systems that can accommodate, and adapt to, their reality rather than our hopes and desires.



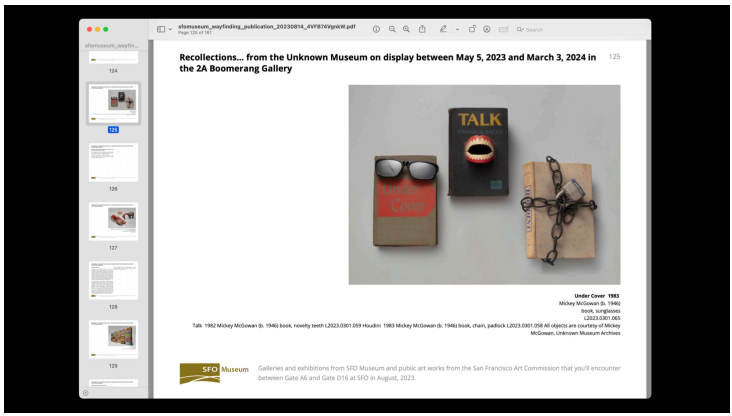
At the top of every page is a **"Create a custom publication for this route"** `https://millsfield.sfomuseum.org/blog/2023/06/12/wayfinding-pdf/` link.



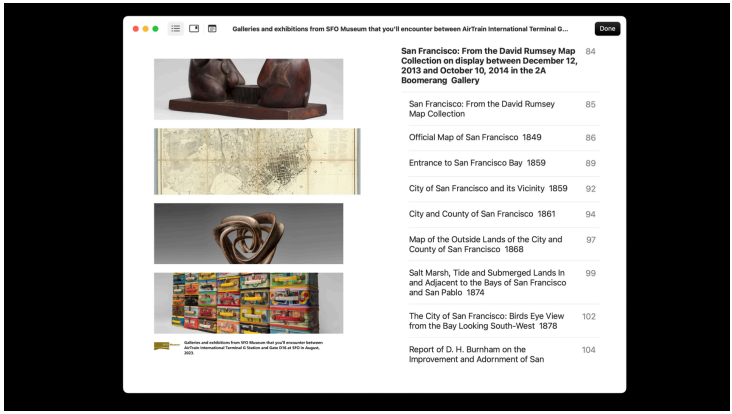
Custom publications are PDF or EPUB "books" that are generated on-demand and contain maps and all the Museum-related programming along a route. You can also choose to include some of, or all, the past exhibitions for galleries along a route.



Here's a screenshot of what the title page for a PDF publication looks like, complete with a map and the wayfinding route. These are formatted for standard US letter-sized pages with the idea that they might be printed out for things like a school field trip to the airport or for passengers who know, in advance, that they will have a long layover at SFO.

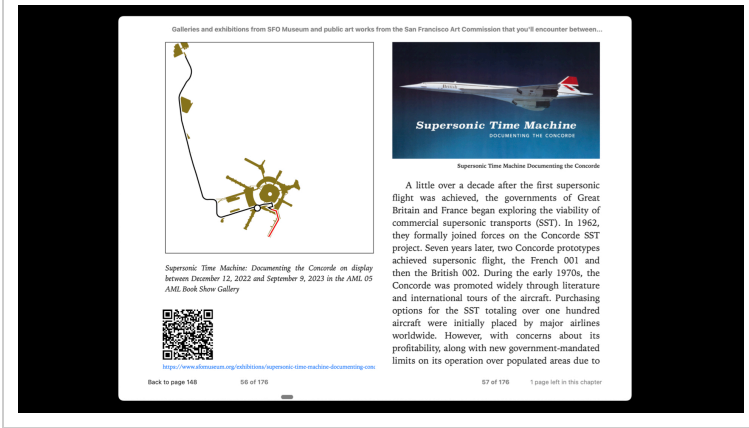


Here's an example of what an object page for a PDF publications looks like. The Museum produces a number of studio-quality photographs for every exhibition each of which is an opportunity to have something nice enough that you'd want to print it out and pin it to your wall.



Here is a screenshot of what the table of contents for an EPUB publication looks like. If PDF files are better suited to print hardcopies of a publication then **EPUBs are better for smaller screens** `https://millsfield.sfomuseum.org/blog/2023/07/10/wayfinding-epub/` and the kinds of mobile devices that a person might take on a flight with them. We would like nothing more than to produce an on-demand publication measuring hundreds of pages in length and spanning decades worth of exhibitions that someone will read at 35,000 feet.

For example, in this screenshot you can see the **exhibition of maps from the David Rumsey Collection** `https://www.sfomuseum.org/exhibitions/san-francisco-david-rumsey-map-collection` that was on display in the "Boomerang Gallery" in Terminal 2 in 2012.



And here is a screenshot of the title page for a given exhibition in an EPUB publication. An online, digital wayfinding system is a useful tool to help situate passengers in the moment. This system and the publications we can derive from it are a useful device to help develop and reenforce a spatial awareness of the totality of the terminal complex and everything it has to offer, notably the museum.

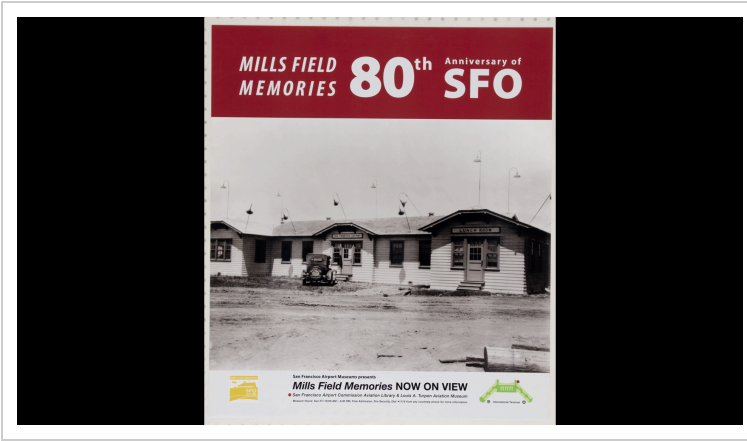
But one of the important threads that runs through all of this work is the idea that visiting the Museum, that fundamentally the Museum itself, extends beyond the building.

3D postcard: Japan Air Lines, Boeing 727, Tokyo International Airport *1960s*
Gift of Thomas G. Dragges
Collection of SFO Museum ***2015.166.0822***

That simple idea is also largely antithetical to the way most museums operate in 2023.

Museums are desperate to get you in the door and to keep you there for as long as they possibly can because, while they will never admit to this, most of them have no expectation you will ever come back. Similarly, an airport has some obvious economic incentives for getting you to arrive earlier than your flight's departure time but sooner or later we know you will leave. In fact, we expect you to leave.

We also know that many, if not most, of those passengers will return to SFO for the second half of their trip if not for an entirely new trip in the future. The great advantage that a museum in an airport has is that we enjoy repeat visitation by default.



Exhibition poster: San Francisco Airport Museums, Mills Field Memories *2007*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2017.079.032***

It's not really considered good form, in museum circles, to say this out loud but the reality is that in 2023 few museums enjoy any kind of serious repeat visitation. Most people will ever only visit a given museum once. If they are repeat visitors to a museum fewer of them, still, will be repeat visitors to any given exhibition let alone a single object.

Before COVID-19 this may not have seemed like a problem because the sheer volume of tourists pouring in to museums for selfies or to check things off their bucket lists was sufficiently large that it was easy to ignore the more fundamental questions that repeat visitation, or the lack of it, pose.

Fun fact: The buildings in this photograph are a runway now.

Toy set: "Sky Rangers" lighthouse tower *1940s*
Purchase
Collection of SFO Museum ***1995.29.18 a d***

I think these questions are important because I believe the act of revisiting is the bedrock of cultural heritage and the humanities.

I believe that the thing which distinguishes entertainment from culture is the act of revisiting. There is a time and a place for both things and, importantly, works which begin life as "entertainment" routinely transition to become "culture". But it is precisely in the act of revisiting these works of entertainment, though, that their transition to "culture" occurs.

It is in the turning over of a work, of debating it and viewing it from different vantage points, again and again and again, that something becomes more meaningful than simple escapism or **the stimulating of nerve-endings** `https://www.aaronland.info/weblog/2014/09/11/brand/#dconstruct` .

I want to argue that contemporary museums have evolved to make revisiting the exhibitions they show and the collections they shepherd as difficult and prohibitive as possible. This happens in a number of ways but chief among them are the prohibitive costs of entry. The economics of running a museum in the early 21st century are a hot mess, for a lot of complicated reasons, *but all visitors know is that it costs them upwards of 30$ each time they want to enter a museum*.

That's a pretty powerful disincentive for most people and one that is compounded further by the inability or the unwillingness of museums to meaningfully embrace the internet as a tool to make any kind of post-visit (or even pre-visit) interactions possible.

Negative: San Francisco International Airport (SFO), United Air Lines, Douglas DC-8-50 *1964*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2011.032.1256***

If a museum's purpose is not to foster revisiting, though, then how is it different from any other manufactured distraction available to the public today?

If museums, by their actions, actively work against the very idea of revisiting as anything but a commercial transaction why then should they enjoy a position of prestige relative to video games, for example, or other media companies? Why should anyone care about all these "weird, old things" given all the other options?

I want to be clear about something: I do not think that SFO Museum's Wayfinding system, alone, will solve these problems. The museum sector, as a whole, has painted itself in to a pretty deep corner and I am not sure it's accepted that reality yet let alone

thought about ways to address the issues it faces. This is the burden of museum professionals working in 2023.

Photograph: San Francisco Airport, model *1936*
Transfer from San Francisco International Airport
Collection of SFO Museum ***1997.52.064.017***

The SFO Museum Wayfinding system is one attempt to build a practical solution to a specific problem – Where is something from the Museum at the airport relative to a passenger and how do we direct that passenger there? – and to do so within the operational and financial constraints of a museum.

But it is also an attempt to make manifest, in both theory and practice, **the idea that a museum at an airport can be a place of revisiting** `https://www.aaronland.info/weblog/2015/11/09/keinholz/#mcn` . That the value of **the Wayne Thiebaud painting in Terminal 3** `https://flickr.com/photos/sfac/13316264535` , for example, comes from being able to spend 10 minutes with it before your flight *every time you come to the airport*. And that the museum will follow you out of the airport – through the use of wayfinding tools and custom publications and other devices – as a way to ackowledge people's busy lives and as a way to encourage

them to revisit something they might have missed the last time they were at SFO.

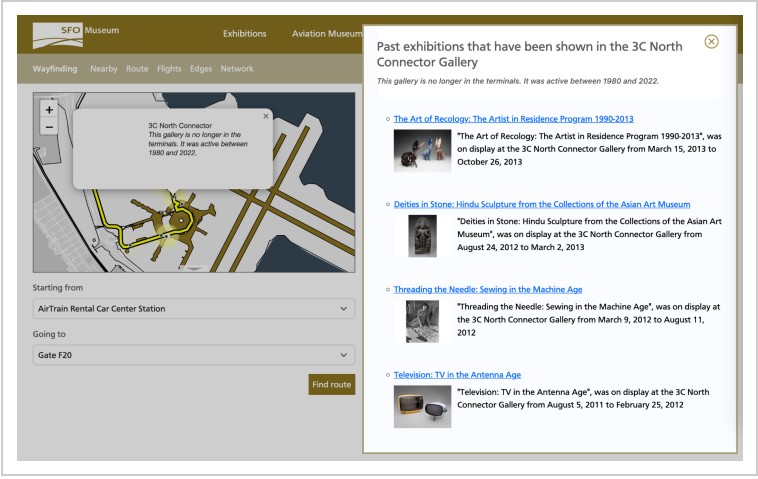
Thank you.

2023-08-21

things I have written about elsewhere #20230912

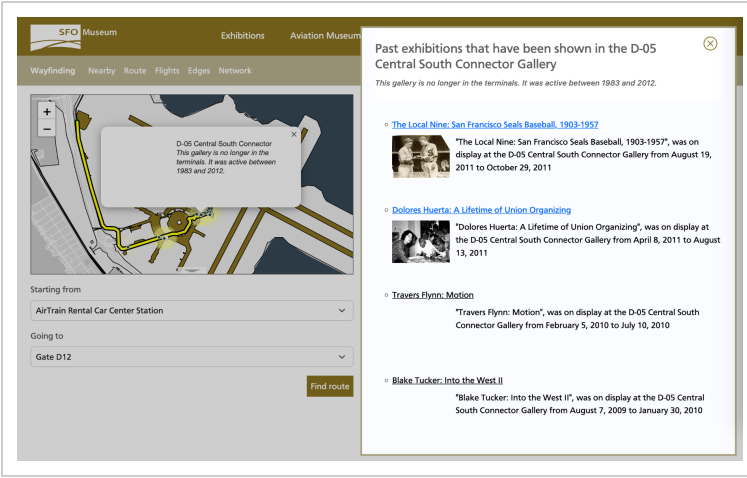
Updating the SFO Museum Wayfinding Service -...

Updating the SFO Museum Wayfinding Service - Past Galleries (and Exhibitions) Along a Route

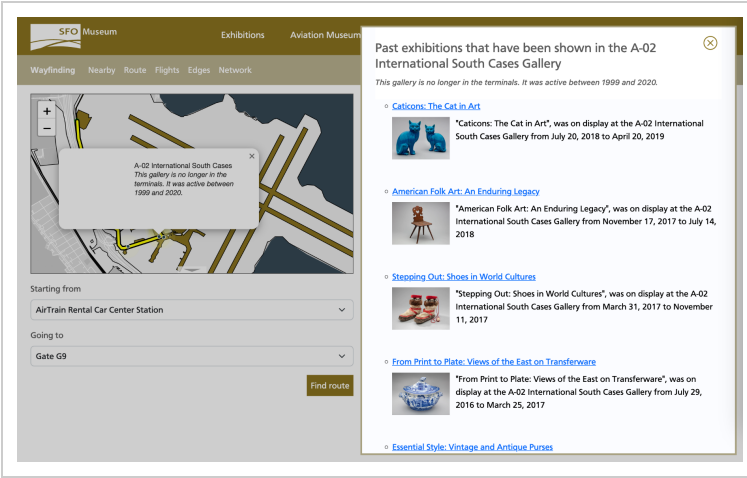


This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/09/12/wayfinding-past-galleries/` , in September 2023.

This is a very short post to mention that the **SFO Museum Wayfinding system** `https://millsfield.sfomuseum.org/wayfinding/` has been updated to display past galleries along a route as well as all the past exhibitions in those past galleries, notably the "3C North Connector" gallery in Boarding Area F **which many people believe is the sum total of the museum** `https://millsfield.sfomuseum.org/blog/2023/08/16/wayfinding-past/` but which closed last year.



Not all past exhibitions in past galleries displayed along a route in the SFO Museum Wayfinding system have studio-quality photographs of their objects but for many of those exhibitions we have "installation" photos and those will be integrated in to the Wayfinding system shortly. In the meantime here is a blog post titled **"People Looking at Art at SFO (1982 - 2019)"** `https://millsfield.sfomuseum.org/blog/2019/02/26/people/` of awesome installation photos taken over the years.



Finally here is the link to **a talk I recently did about the SFO Museum Wayfinding system** `https://www.aaronland.info/weblog/2023/08/21/quiet/#wayfinding` and why I think it and especially things like past galleries and exhibitions are important for museums:

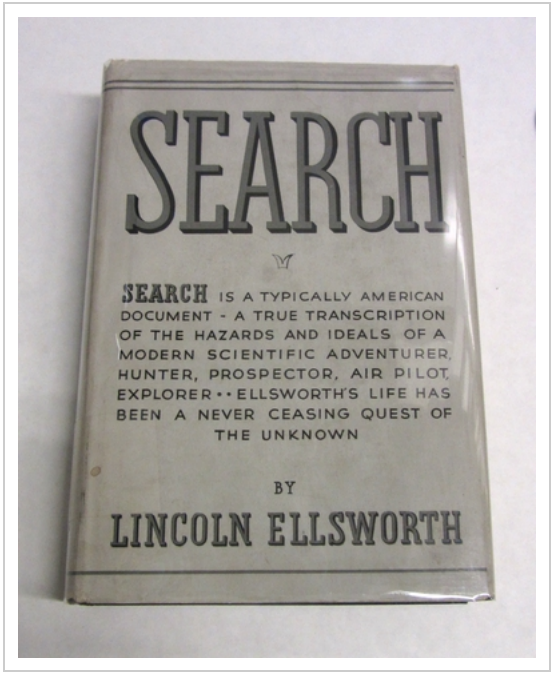
https://www.aaronland.info/weblog/2023/08/21/quiet/#wayfinding `https://www.aaronland.info/weblog/2023/08/21/quiet/#wayfinding`

2023-09-12

things I have written about elsewhere #20230914

Searching Text in Images on the Aviation...

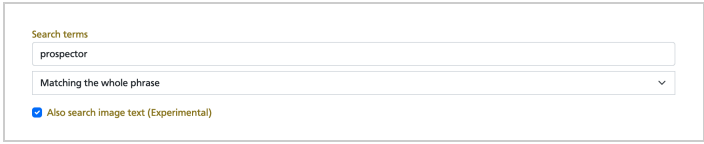
Searching Text in Images on the Aviation Collection Website



Search. Paper, ink. The Tony Bill Aviation Library Collection, SFO Museum Collection. ***2000.160.0319***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/09/14/image-text-search/` , in September 2023.

We are pleased to announce the introduction of a new, experimental, feature for search on the **SFO Museum Aviation Collection website** `https://collection.sfomuseum.org` : The ability to search for **text contained *in the images* of objects** `https://collection.sfomuseum.org/search/advanced/?q=prospector&mode=phrase&search_image_text=true` as well as the metadata (titles, descriptions, etc.) for those objects.



Because it is still an experimental feature there are some caveats that apply:

1. ***It is only available from the advanced search*** `https://collection.sfomuseum.org/search/advanced/` ***page and you must explicitly toggle a checkbox indicating you want to also search for a query term in the text that has been extracted from images.***

2. ***While the tools we're using to extract text from images are very good*** `https://developer.apple.com/documentation/vision/` ***they still don't recognize all text, don't necessarily know to recognize phrases, and sometimes, in what we'll assume is a well-meaning attempt to be helpful, simply return made up and gibberish text.***

3. ***There are whole other conversations to be had about the machine learning processes being used behind the scenes that make this kind of functionality possible. These conversations are not the subject of this blog post but it is important to call attention to them and be explicit about the risks and open questions they pose*** `https://www.wired.com/story/google-timnit-gebru-ai-what-really-happened/` ***and is one reason this work is considered "experimental" and why it is not enabled by default.***

Airline bag: Canadian Pacific Airlines. Synthetic textile, plastic, metal, fabric, paint. Gift of Thomas G. Dragges, SFO Museum Collection. ***2001.146.091***

We want to see what these technologies make possible, though, and a very real and immediate opportunity is the ability to index search terms that the curators and registrars haven't already included in the titles or descriptions for objects in our collection. For example, this **travel bag from Canadian Pacific Airlines** `https://collection.sfomuseum.org/objects/1763308157/`, covered in the names of cities the airline traveled to, is not included in the search results for the terms `montreal` or `saskatoon` using the default search functionality but it's the first result when we **also search for the text in images** `https://collection.sfomuseum.org/search/advanced/?q=saskatoon&mode=phrase&search_image_text=true` .

There are **four images of this travel bag** `https://collection.sfomuseum.org/objects/1763308157/images/` and these are the texts that we've extracted from those images:

LOS ANGELES [1 VICTORIA (NUR EDMONTON (WINNIPEO RIcani [ISASKATOON STORONTO IRALA SANTIAGO [SI HONOLULU (AUSTERA VANCOUVER MOTTAWA (EDMONTO BUENOS AIRES ((SASKATOON [(us80l SAN FRANCISCO (MILAN (HONGKON (WINNIPEG (VANCOUVER LICALCART OTTAWA (LOSANGELES & HONOLUL) KvANcOUVe

LOSINGELES A VICTORIA I HALIFAS DIDNTO L WINNIPEG I SANTIAGO (GASLITION STORONTO A HALIFAX MITIAGO & HONOLULU I AMSTERDAM KIANCOUVER lOTTAWA EDMONTON BUENOS AIRES I SASKATOON A LISBON LAN FRANCISCO MILAN I HONG KONG (WINNIPEG (VANCOUVER I CALGARY OTTAWA I LOS ANGELES I HONOLULU VANCOUVER

TORTO I SASKATOON ROME I MILAN I VICTORIA MAN) SANTIAGO I EDMONTON I REONA I MONTREAL I SANTIAGO LISBON SAN FRANCISCO E(SYONEY L ALAN K/ MILAN IRONE COrONEr KIROM MILAN I BUENOS ARES Snowman) LOS ANGELES AUCTORIA «(aura) SANTIAGO [I HONOLULU LA ANSTEROL BUENOS AIRES SABRATOON LAus SAN FRANCISCO MILAN (ON NO! •WINNIPEG IVANCOUVER LIONEL OT

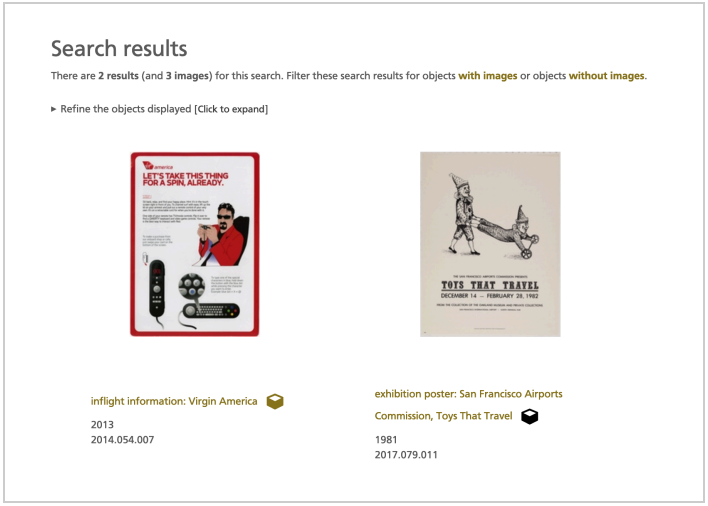
TOKTO KI SASKATOON ROME I MILAN I VICTORIA PI PI) MITIAGO I EDMONTON I REGINA HONTREAL [I SANTIAGO C LISBON MI FRANCISCO (I SYDNEY I MILAN MILAN I ROME I SYDNEY (ROME TN LI BUENOS AIRES I MONTREAL MANGELES I VICTORIA I MALIFAX BRONTON E WINNIPEG SI SANTIAGO (SBASSATOON 2 TORONTO C HALIFAX SANTAGO L HONOLULU C AMSTERDAM SIANCOUVER LOTTAWA E EDMONTON BUSIOSARES ASASKATOON LISBON SAN FRANCISCO I MILAN HONG KON° [(WIMPEO

[IVANCOUVER (CALGARY

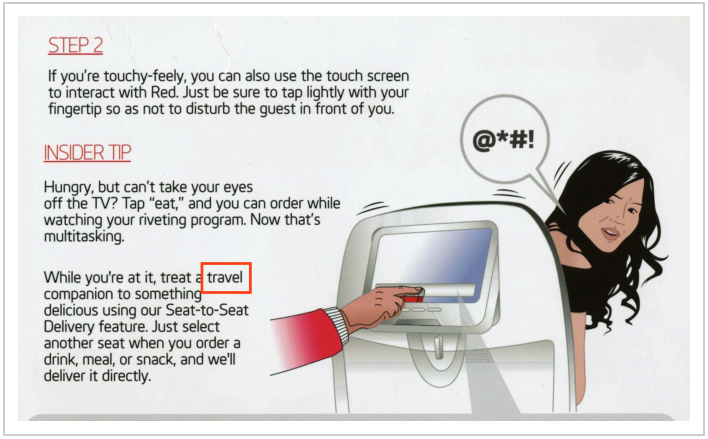
It's clear that there are still mistakes (`LOSINGELES`, `SABRATOON`, `MALIFAX`, `SIANCOUVER`, `LAN FRANCISCO` and so on) but there are also just as many correct place names and that's a nice addition to the descriptive text which accompanies the object:

> Airline bag issued by Canadian Pacific Airlines, manufactured by TQ Tradex; brown background with dark brown piping, red logo and various city destinations as pattern on front pocket and back; adjustable shoulder strap with handles and top zipper closure; four plastic feet at bottom.

There is still a lot of experimenting to do to better understand the affordances and tolerances of searching for text in images. If I search for **the terms "toys" and "travel" matching on individual words** `https://collection.sfomuseum.org/search/advanced/?q=toys+travel&mode=string&search_image_text=true` (rather than a complete phrase) I get back two results:



One is **a poster for the "Toys that Travel" exhibition** `https://collection.sfomuseum.org/objects/1779445165/` from 1982, on the right, and it's easy to see why that was included (both terms are also included in the object's descriptive text). The other is an **in-flight information card from Virgin America** `https://collection.sfomuseum.org/objects/1880235993/`, on the left, and it's not immediately obvious why it's been included.



In fact, both search terms appear on **the second photograph for the object** `https://collection.sfomuseum.org/objects/1880235993/#1880236133` depicting the backside of the information card and we currently don't convey that information anywhere in the search results.

One reason we don't convey these details is because the tools we're using don't capture information about where individual words and phrases are located in the images we're scanning so that's an obvious next step to investigate. Those tools are described below and have been **open-sourced** `https://github.com/sfomuseum?q=swift-text-emboss` and we would welcome any feedback or help to improve them because we think the ability to effectively index search terms from images is generally useful for all museums.

In the meantime we hope that the ability to search for text in images will open up the collection to **queries that might have yielded no results** `https://collection.sfomuseum.org/search/advanced/?q=opium&mode=phrase&search_image_text=true` and, more generally, act as new avenue to discover objects in our collection that might have otherwise been missed.

Next steps

Airmail flight cover: Pan American World Airways, Polar Route, Seattle- London. Paper, ink. Gift of Captain Jon Simmonds, SFO Museum Collection. ***2014.156.211***

Other next steps include thinking about the design and interface elements

that would allow visitors to search for text in images directly from the search box at the top of every page and creating a new workflow for **recognizing and extracting structured addresses** https://developer.apple.com/documentation/vision/recognizing_text_in_images from the over **1,600 airmail flight covers** https://collection.sfomuseum.org/objects/subcategories/1511213159/with-images/ in the collection (and to **geocode all those addresses** https://millsfield.sfomuseum.org/blog/2019/11/04/placeholder/ afterwards). There's no timeline for any of this yet but we have started to think about it all.

The rest of this blog post gets in to the plumbing of how we are extracting text from images. If that's not of interest you can stop reading now and we invite you to investigate the new searching-text-in-images feature on the advanced search webpage:

https://collection.sfomuseum.org/search/advanced/ https://collection.sfomuseum.org/search/advanced/

Under the hood

Aviator's goggles. Plastic, metal, canvas, fur, elastic. Gift of Charles Page, SFO Museum Collection. ***2010.174.002***

We are using Apple's **Vision Framework** `https://developer.apple.com/documentation/vision` to extract text from images and we have written a number of tools to make using the Vision Framework a little easier. The first is **swift-text-emboss** `https://github.com/sfomuseum/swift-text-emboss` which I wrote about in **the blog post about the Accession Numbers Project** `https://millsfield.sfomuseum.org/blog/2021/12/30/accession-numbers/` and is really just a helper library hiding the details of working with Apple's underlying **VNRecognizeTextRequest** `https://developer.apple.com/documentation/vision/vnrecognizetextrequest` methods. Here's an abbreviated version of what that looks like:

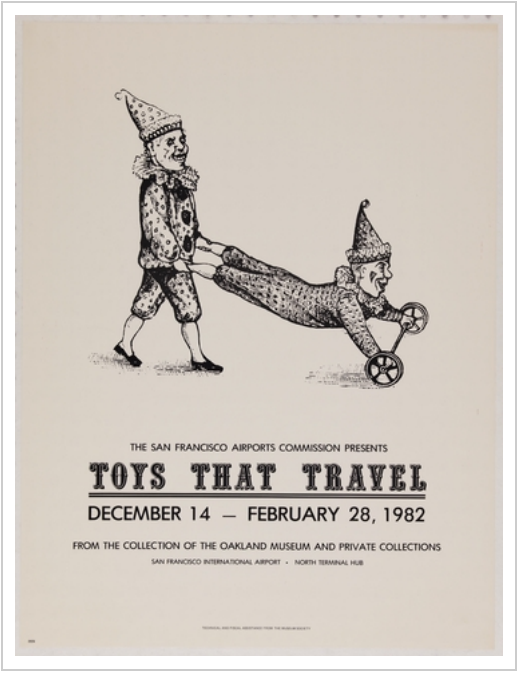
```
import Foundation
import AppKit
import TextEmboss

let im = NSImage(byReferencingFile:path_to_image)
let cgImage = im.cgImage(forProposedRect: nil, context: nil, hints: nil)

let te = TextEmboss()
let rsp = te.ProcessImage(image: cgImage)
```

```
if case .success(let txt) = rsp {
        print(txt)
}
```

Name pin: United Air Lines, Sandy Herrmann. Metal, ink. Gift of Sandra Herrmann, SFO Museum Collection. ***2019.081.021***

The second tool is **swift-text-emboss** `https://github.com/sfomuseum/swift-text-emboss-cli` which is a command-line tool that uses the `swift-text-emboss` package. It accepts the path to an image, extracts the text in that image and then prints the result to `STDOUT`. For example:

```
$> text-emboss sfomuseum-pin-2019.081.021.jpg
SANDY HERRMANN
```



Exhibition poster: San Francisco Airports Commission, Toys That Travel. Paper, ink. Transfer from San Francisco International Airport, SFO Museum Collection. ***2017.079.011***

The third tool is **swift-text-emboss-www** `https://github.com/sfomuseum/swift-text-emboss-www` and is similar to `swift-text-emboss-cli` but instead of providing a command-line interface to extract text from images it starts an HTTP server that accepts image uploads and returns any text that's been extracted (from the uploaded image) in the body of the response. For example, to start the server you'd do this:

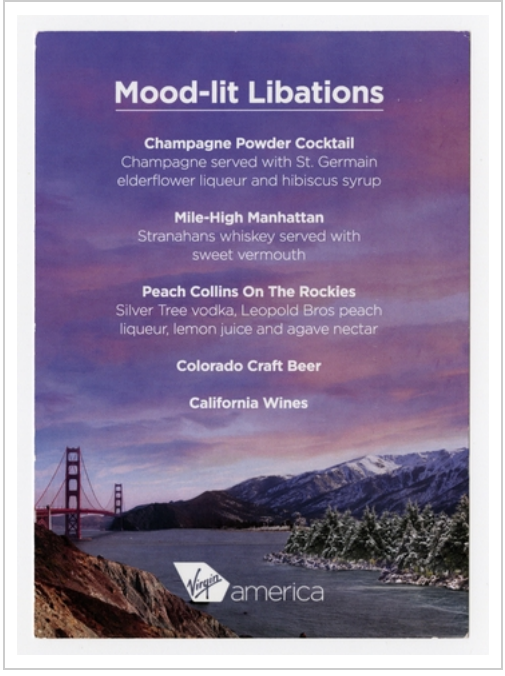
```
$> text-emboss-server
2023-08-31T11:36:32-0700 info org.sfomuseum.text-emboss-server : [text_emboss_server] Ser
```

And then in another window you would do something like this:

```
$> curl -F image=@1779445839_SfgRA4d51gAzgbClRnghkRVINAw2rOjF_b.jpg http://localhost:8080
THE SAN FRANCISCO AIRPORTS COMMISSION PRESENTS
TOYS THAT TRAVEL
DECEMBER 14 –
– FEBRUARY 28, 1982
```

FROM THE COLLECTION OF THE OAKLAND MUSEUM AND PRIVATE COLLECTIONS
SAN FRANCO INTERNATIONAL AIRPORT • NORTH TERMINAL STIR

We wrote `swift-text-emboss-www` to account for the fact that we do nearly all the work to export data and images for the **collection.sfomuseum.org** `https://collection.sfomuseum.org` website on Linux machines. The `Vision` framework is only available on the MacOS and iOS platforms so the idea is that we can continue to do most of our work on Linux machines but when it comes time to extract text from images we can call the `swift-text-emboss-www` server running on a dedicated MacOS machine. In our case that will be a spare laptop that is no longer being used for day-to-day work but still has enough life left in it to be useful for extracting text from images instead of just becoming a paperweight.



Menu: Virgin America. Paper, ink. Gift of Faye Jose, SFO Museum Collection. ***2018.157.039***

The fourth tool is **swift-text-emboss-grpc** `https://github.com/sfomuseum/swift-text-emboss-grpc` and is similar to `swift-text-emboss-www` except that instead of exposing text-extraction services over HTTP (the web) it uses the **gRPC** `https://grpc.io/` protocol which is a stricter and generally more efficient way of doing the same thing (uploading an image and receiving the text contained in it). For example, to start the server you'd do this:

```
$> text-emboss-grpc-server
2023-09-01T11:48:13-0700 info org.sfomuseum.text-emboss-grpc-server : [text_emboss_grpc_s
```

And then using **this client** `https://github.com/sfomuseum/go-text-emboss#remote-grpc` in another window you'd do this:

```
$> go run cmd/emboss/main.go -embosser-uri 'grpc://localhost:1234' menu.jpg
Mood-lit Libations
Champagne Powder Cocktail
Champagne served with St. Germain
elderflower liqueur and hibiscus syrup
Mile-High Manhattan
Stranahans whiskey served with
sweet vermouth
Peach Collins On The Rockies
Silver Tree vodka, Leopold Bros peach
liqueur, lemon juice and agave nectar
Colorado Craft Beer
California Wines
"america
```

Importantly, there is nothing special about **the gRPC client** `https://github.com/sfomuseum/go-text-emboss#remote-grpc` we are using in this example. The whole promise of gRPC is that **it enjoys broad support across a number of programming languages** `https://grpc.io/docs/languages/` and because it uses **formal definitions** `https://github.com/sfomuseum/swift-text-emboss-grpc/blob/main/Sources/text-emboss-grpc-server/embosser.proto` for its input and output it is very easy to create your own custom gRPC client for any given service.

Honestly, I wasn't planning on writing a gRPC server but I had enough time to wait while running a backfill process that I decided to do it as a kind of **wax on, wax off** `https://www.youtube.com/watch?`

`v=Bg21M2zwG9Q` exercise and, in the end, this is probably what we'll use going forward. It is not orders of magnitude faster than the HTTP server but it is faster. It is also a good template for other services taking advantage of Apple-specific libraries and frameworks that I want to investigate in the future.

Beverage server: Transcontinental & Western Air (TWA). Metal, paint. Gift of the San Francisco Aeronautical Society, SFO Museum Collection. ***2022.060.063***

All of these tools are available from **SFO Museum's GitHub account** `https://github.com/sfomuseum` and we welcome any feedback, suggestions or patches that you think would help to improve them:

1. ***https://github.com/sfomuseum/swift-text-emboss*** `https://github.com/sfomuseum/swift-text-`

`emboss`

2. ***https://github.com/sfomuseum/swift-text-emboss-cli*** `https://github.com/sfomuseum/swift-text-emboss-cli`

3. ***https://github.com/sfomuseum/swift-text-emboss-www*** `https://github.com/sfomuseum/swift-text-emboss-www`

4. ***https://github.com/sfomuseum/swift-text-emboss-grpc*** `https://github.com/sfomuseum/swift-text-emboss-grpc`

Finally, the text extracted from images is included in the data contained in both the **sfomuseum-data-media-collection** `https://github.com/sfomuseum-data/sfomuseum-data-media-collection` and **sfomuseum-data-collection** `https://github.com/sfomuseum-data/sfomuseum-data-collection` repositories, available from the **sfomuseum-data GitHub organization** `https://github.com/sfomuseum-data` .

2023-09-14

things I have written about elsewhere #20231031

Extracting Subjects from Images in Swift (and...

Extracting Subjects from Images in Swift (and gRPC)

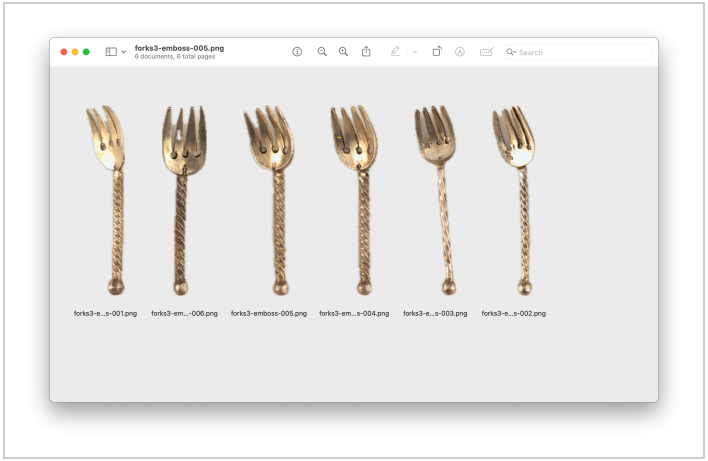
Flight attendant scarf: National Airlines. Polyester. , SFO Museum Collection. ***2016.138.001***

In the **Searching Text in Images on the Aviation Collection Website** `https://millsfield.sfomuseum.org/blog/2023/09/14/image-text-search/` blog post I introduced the **swift-text-emboss** `https://github.com/sfomuseum/swift-text-emboss` Swift package, a wrapper library around Apple's **Vision Framework** `https://developer.apple.com/documentation/vision` to simplify extracting text from images. In this blog post I'd like to introduce the **swift-image-emboss** `https://github.com/sfomuseum/swift-image-emboss` Swift package. Like the `swift-text-emboss` package this is also a wrapper around code provided by Apple's Vision Framework with the goal simplifying the code necessary to extract, or "lift", one or more subjects from an image.

It works like this:

```
import ImageEmboss

// This is the image you want to extract subjects from.
var ciImage: CIImage

let em = ImageEmboss()
let rsp = em.ProcessImage(image: ciImage, combined: false)
```

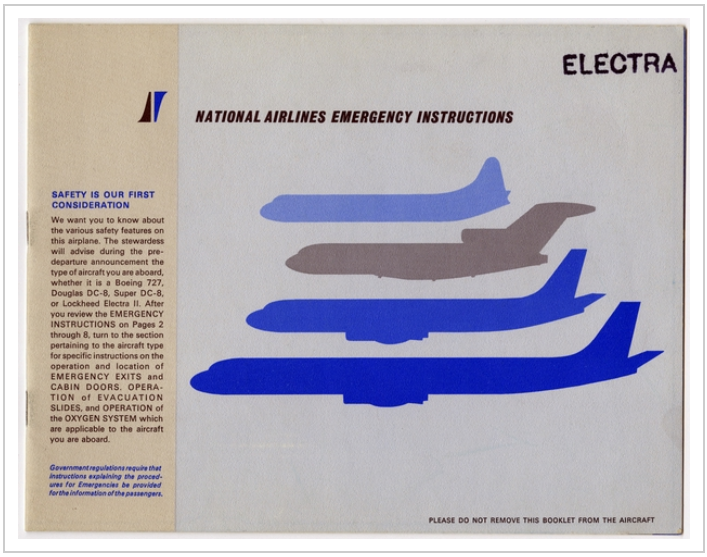
The `rsp` variable is a Swift **Result** `https://developer.apple.com/documentation/swift/result` instance wrapping a list of `CIImage` instances or any errors encountered during processing. For example, when "embossed" this image of **six forks from the Cooper Hewitt Smithsonian National Design Museum** `https://collection.cooperhewitt.org/objects/18381819/` yields six individual images where each fork is considered a subject:



To extract all the subjects in an image in to a single image you would call the `ProcessImage` method passing in a `combined: true` argument. For example, this image of **the items in an Air France amenity bag** `https://collection.sfomuseum.org/objects/1780469983/` when processed in "combined" mode produces a single image with five distinct subjects:

Note that the underlying "subject lifting" code merges two objects in to one and excludes four others entirely. This is a good reminder that as impressive as the technologies used to perform these tasks are they remain

fallible and the reasons they choose to do, or not do, something often remain opaque. While they herald exciting new possibilities they are still tools which should only be used with supervision and a good understanding (or at least awareness) of when and where they may not meet the needs of your specific project.



Safety information card: National Airlines, Lockheed L-188 Electra II. Paper, ink. Gift of Thomas G. Dragges, SFO Museum Collection. ***2002.127.425***

The `swift-image-emboss` package provides a library that you can include in your own code but does not export any applications of its own. We have released a separate tool, called **swift-image-emboss-cli** `https://github.com/sfomuseum/swift-image-emboss-cli`, for extracting subjects from images on the command line. For example, processing **this image of an airplane resting atop a walrus** `https://collection.sfomuseum.org/objects/1511908311/` :

```
$> cd swift-image-emboss-cli
$> swift build

$> ./.build/debug/image-emboss --input-file fixtures/sfomuseum-walrus-1511908311.jpg
```

*This is really a thing in the **SFO Museum Aviation Collection*** `https://collection.sfomuseum.org` *!*

We have also released a **gRPC** `https://grpc.io/` server, called **swift-image-emboss-grpc** `https://github.com/sfomuseum/swift-image-emboss-grpc`, wrapping the `swift-image-emboss` functionality so that you can extract subjects from images over the network. For example:

```
$> cd swift-image-emboss-grpc
$> swift build

$> ./.build/debug/image-emboss-grpc-server
2023-10-26T14:13:25-0700 info org.sfomuseum.text-emboss-grpc-server : [GRPCServer] server
```

And then using a gRPC client, for example the client provided by the **go-image-emboss** `https://github.com/sfomuseum/go-image-emboss`

package, to extract subjects from this image of **a tie tack from Civil Air Transport (CAT) airlines** `https://collection.sfomuseum.org/objects/1762759391/` ::

```
$> cd go-images-emboss
$> go build -mod vendor -ldflags="-s -w" -o bin/emboss cmd/emboss/main.go

$> ./bin/emboss -embosser-uri 'grpc://localhost:8080' fixtures/cat-pin.jpg
2023/10/26 14:15:24 fixtures/cat-pin-emboss-001.png
```

Results in:

A couple of things to be aware of:

First, there is nothing special about **the gRPC client** `https://github.com/sfomuseum/go-image-emboss#remote-grpc` we are using in this example. The whole promise of gRPC is that **it enjoys broad support across a number of programming languages** `https://grpc.io/docs/languages/` and because it uses **formal definitions** `https://github.com/sfomuseum/swift-image-emboss-grpc/blob/main/Sources/ImageEmbossGRPC/embosser.proto` for its input and output it is very easy to create your own custom gRPC client for any given service. Second, **the protocol definition** `https://github.com/sfomuseum/swift-image-emboss-grpc/blob/main/Sources/ImageEmbossGRPC/embosser.proto` for the `swift-image-emboss-grpc` server only accepts and returns raw (image) bytes so it is left to inidividual applications (like the `go-image-emboss` tool above) to determine how those bytes should be read from and written to.

Negative: San Francisco International Airport (SFO), runway construction. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.0518***

The `image-emboss-grpc-server` gRPC server implementation was sufficiently similar to the equivalent "text embosser" server **described in the last blog post** `https://millsfield.sfomuseum.org/blog/2023/09/14/image-text-search/` that we created a standalone **swift-grpc-server** `https://github.com/sfomuseum/swift-grpc-server` Swift package which both the image and text "embosser" servers now use. It works like this:

```
import Logging
import GRPCServer
import GRPC

// This is your own code, for example the image embosser or the text embosser.
// This is the package that will be created when you run protoc on your protobuffer
// definition. Consult the swift-grpc docs for details.
var provider: CallHandlerProvider

let logger = Logger(label: "org.sfomuseum.example")

let server_opts = GRPCServerOptions(
        host: "localhost",
        port: 8080,
        logger: logger
)

let server = GRPCServer(server_opts)
try await server.Run([provider])
```

The goal with the `GRPCServer` package is to reduce the amount of boilerplate code necessary for simple gRPC servers that need to log requests (and in particular record the remote address of clients connecting to the server **which is more complicated than you'd imagine** `https://github.com/sfomuseum/swift-grpc-server#logging-remote-address` and, optionally, **use TLS to secure connections and provide authentication** `https://github.com/grpc/grpc-swift/blob/main/docs/tls.md` . It is the minimal amount of scaffolding necessary to expose functionality not available on other platforms as a network-based service.

Timetable: Southwest Airlines. Paper, ink. Gift of Craig Lynes, SFO Museum Collection. ***2003.105.097.005***

We've also abstracted one other piece of code in to its own dedicated software package: **swift-coreimage-image** `#` . SFO Museum often writes tools in Swift that we would like to run in both MacOS (**AppKit** `https://developer.apple.com/documentation/appkit/`) and iOS (**UIKit** `https://developer.apple.com/documentation/uikit`) contexts. Because each context has its own image type (`NSImage` and `UIImage` respectively) and each framework is specific to their respective platform this can result in a lot of fiddly-code when cross-compiling applications.

This package provides a single method (`LoadFromURL`) which accepts a **URL** `https://developer.apple.com/documentation/foundation/url` instance and returns a **CoreImage**

CIImage `https://developer.apple.com/documentation/coreimage/ciimage` instance for the image at that location, regardless of platform. That's all it does. How many people have already written this code? I'd wager a lot of people have but I couldn't find any of those efforts in a simple package that we could import in to our code so now we have our own implementation which we are sharing with you.

```
import CoreimageImage

let im_url = URL(string: "file:///path/to/image.jpg")
let im_rsp = CoreImageImage.LoadFromURL(url: im_url)

if case .success(let ci_image) = im_rsp {
    // do something with ci_image here
}
```

Load adjuster: Pan American Airways, Stratocruiser. Leather, wood, plastic, metal, ink. Gift of Jon E. Krupnick, SFO Museum Collection. ***2022.124.782***

The `swift-image-emboss` package, and the layering of tooling on top of it, were written specifically with the goal of improving the quality of results deriving dominant colors in images from the aviation collection, by isolating and focusing on the principal subjects in those images rather than the entire frame. That is SFO Museum's use-case but as part of our

commitment to the idea of "**small focused tools** `https://millsfield.sfomuseum.org/blog/2020/08/04/small-tools/` " by and for the cultural heritage sector we have made the effort to package as discrete reusable (and rearrange-able) components for use in a variety of circumstances.

> The cultural heritage sector needs as many small, focused tools as it can produce. It needs them in the long-term to finally reach the goal of a common infrastructure that can be employed sector-wide. It needs them in the short-term to develop the skill and the practice required to make those tools successful. We need to learn how to scope the purpose of and our expectations of any single tool so that we can be generous of, and learn from, the inevitable missteps and false starts that will occur along the way.

All of these tools are available from **SFO Museum's GitHub account** `https://github.com/sfomuseum` and we welcome any feedback, suggestions or patches that you think would help to improve them:

1. ***https://github.com/sfomuseum/swift-image-emboss*** `https://github.com/sfomuseum/swift-image-emboss`

2. ***https://github.com/sfomuseum/swift-image-emboss-grpc*** `https://github.com/sfomuseum/swift-image-emboss-grpc`

3. ***https://github.com/sfomuseum/swift-grpc-server*** `https://github.com/sfomuseum/swift-grpc-server`

4. ***https://github.com/sfomuseum/go-image-emboss*** `https://github.com/sfomuseum/go-image-emboss`

5. ***https://github.com/sfomuseum/swift-coreimage-image*** `https://github.com/sfomuseum/swift-coreimage-image`

2023-10-31

generative adversarial therapy sessions

Wishful Thinking

Wishful Thinking

Postcard: Finnair, Sud Aviation Caravelle *1970s*
Gift of Thomas G. Dragges
Collection of SFO Museum ***2015.166.0701***

I spoke at the ***2023 Museum Computer Network conference*** `https://mcn.edu/annual-conference/mcn-2023/` *, in Philadelphia, last week and presented a talk titled "****Wishful Thinking – A critical discussion of "extended reality" technologies in the cultural heritage sector*** `https://mcn2023.sched.com/event/1RYPT/wishful-thinking-a-critical-discussion-of-extended-reality-technologies-in-the-cultural-heritage-sector` *". These are my notes for the talk. This is what I was planning to say sticking to these notes closely enough (because the talk was only 15 minutes) that I expected people would notice. I said as much at the beginning of the talk but by the time I reached the second slide I realized that I had neglected to make sure that the display setup allowed me to see my notes. It did not and it was too late to do anything about that so I did the talk from memory. I missed a few points that I wanted to make but, somehow, still managed to capture the gist of it.*

I want to take a moment to acknowledge that we are rounding the corner on the first anniversary of the death of **Effie Kapsalis** `https://en.wikipedia.org/wiki/Effie_Kapsalis` . Many of the people at MCN this year will have known or worked with Effie. For those of you who didn't, her's is a name worth remembering. Effie is probably best known for being the spearhead and the public face of the **Smithsonian's Open Access Initiative** `https://www.si.edu/OpenAccess` but she did so much more during her time and by those efforts she made us all, both individually and as a professional community, better.

Postcard: Finnair, Sud Aviation Caravelle *1970s*
Gift of Thomas G. Dragges
Collection of SFO Museum ***2015.166.0701***

This is a short talk for a big subject. As such it is possible some "old man yelling at the sky" vibes might happen and for that I'd like to apologize in advance and invite you to find me after the talk to discuss the nuances and finer points of the subject.

It's a big subject because what I really want to talk about today is not so much a particular set of technologies but what those technologies represent for the cultural heritage sector.

Flight officer cap badge: West African Airways Corporation *1960s*
Gift of the Captain John B. Russell Family
Collection of SFO Museum ***2012.147.401***

In order to do that I am going to trot out of a trio of elephant-sized ideas and then leave them standing awkwardly in the corner. Each one of these ideas is an entire conference in its their right. We don't have time for that today but it is important to call out these ideas because they are the basis, and the core set of beliefs, from which the rest of this talk extends.

Toy set: "Sky Rangers" lighthouse tower *1940s*
Purchase
Collection of SFO Museum ***1995.29.18 a d***

The first is that the practice of revisiting is the bedrock of the humanities. Revisiting is what distinguishes entertainment from culture.

It is important to recognize that there is a time and a place for both things and that entertainment routinely becomes culture but that transformation occurs precisely in the very act of revisiting.

Seat occupied card: TWA (Trans World Airlines) *c. 1965*
Gift of TWA Clipped Wings International
Collection of SFO Museum ***2002.113.523***

The second is that **recall has always been a power dynamic** `http://www.aaronland.info/weblog/2014/09/11/brand/#dconstruct` .

You see this throughout history: In the right to assembly; In the question of access to basic literacy; The entire notion of a public library; The restrictions around intellectual property; Paywalls and the rights to broadcast or re-broadcast.

Whether it's been through malice or negligence or unintended circumstances the ability to not only see in to the future but see in to the past has not been an opportunity open to all.

Negative: San Francisco International Airport (SFO), instrument flight room *1963*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2011.032.0940***

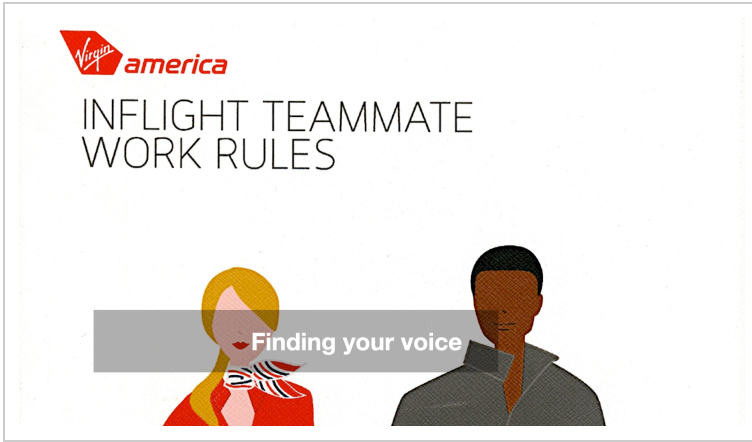
Third, technology is a reflection of worldview. This is a generally true statement, I think, but it is especially true with regards to these last two points. How you think about the role and the function of technology is a pretty good mirror on how you think about the questions of recall and revisiting.

I am going to come back to this idea at the end of the talk because I think it is crucially important to the cultural heritage sector in a moment where we seem to be living through another technological phase-shift.

But first I am going to travel back in time to **Museums and the Web 2013** `https://mw2013.museumsandtheweb.com/` *. (I actually said and believed this was MCN 2013 until I started writing this blog post. Oh well...)* This is a photograph of the inflatable unicorn head that I brought to the **panel discussion on humour and institutional voice** `http://www.aaronland.info/weblog/2013/05/05/design-thanking/#face` that I was a part of.

After the panel I was approached by a recent museum studies graduate who asked for advice on getting started and developing a public persona in the sector. Although they never said these exact words what I heard was:

"How do I as a person who is new to a small and often insular community develop the means to express controversial, or at least unorthodox, opinions without burning all my bridges before I've even gotten started?"



Inflight service manual: Virgin America, work rules *2013*
Gift of Alaska Airlines
Collection of SFO Museum ***2018.083.152***

I do not think that I am especially well-qualified to answer that question. The reality is that I arrived in the museum sector with a reputation, or at least credentials, that preceded me and this was a great privilege not enjoyed by all.

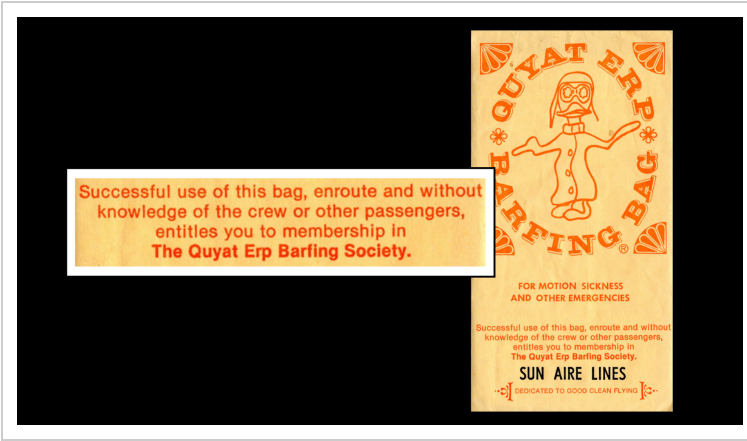
What I tried to explain to this person, though, was that quite a lot of that reputation was made possible by having been present on and participating with the web. Without the web most of my peers and I would never have enjoyed the audiences or the reach for the ideas and projects, some of them serious and many of them goofy, that we pursued.

Importantly, being present on the web was not, and has never been, a guarantee of a receptive audience or even any audience at all.

On the other hand the web made those things possible in a way that

had never existed before. What that means practically is that it was easy and cheap enough to put something on the web and then to leave it there long enough for others to find it, after the fact.

Not free but compared to the limited avenues available to people before the web you could almost pretend that it was.



Airsickness bag: Sun Aire Lines *1970s*
Gift of Denise Harr
Collection of SFO Museum ***2001.115.003 a c***

Already back in 2013, though, you could feel like something was changing. It was a change, one actively advanced and promoted by social media companies in the decade to follow, that's come to largely define how people think about doing anything online, even today.

Namely that if you publish something online – whether it's a selfie, a "hot take", an essay or a multi-year project; anything really – and it is not immediately successful or viral then it was a waste of time and effort. It was not worth doing.

Which is insane. It is insane because that's not how ideas take root.

It takes people time to warm up to ideas, especially new or challenging ideas, if only because we are busy just juggling the ideas and beliefs we already hold with the complexities of our lives in relation to one another.

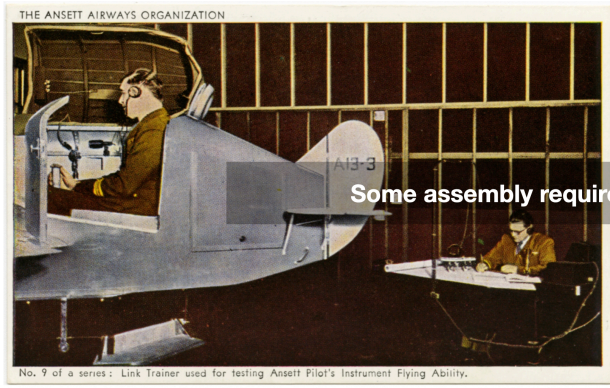
It is also hard not to understand this idea as a deliberate of attempt to gaslight the web and everything that makes the web important.

So let's take a moment to review what makes the web important and novel.

Uniform patch: Rapid Response Worldwide, Seaflite Oceanographic *1970s*
Gift of Frank A. Norick
Collection of SFO Museum ***1999.054.001***

Not only is the web a global network of linked documents unbounded by geography but, critically, it is asynchronous.

Not only can you access the web from anywhere but you can do so at anytime. The technology for these three things to exist in tandem may have pre-dated the web but they had never been realized in concert, at the scale of the entire planet, before the web.



Some assembly required

Postcard: Ansett Airways *1940s*
Gift of Thomas G. Dragges
Collection of SFO Museum ***2015.166.0058***

Both HTML, the format used to encode documents on the web, and HTTP, the mechanism to deliver those documents from a server to a browser, are radically simple in their design.

It is true that, in 2023, the web has evolved to eye-watering degrees of complexity in order to allow for higher-order functionality like streaming video, advanced layouts and fancy animations.

But the core of the web, the design for deploying a network of linked documents, of words and ideas, remains simple enough that most people can create a web page in day and many people could build a rudimentary but functional client or a server in a weekend.

And this. This is a screen capture of the opening ceremony at the London Olympics in 2012.

It might be hard to see but that's a little house in the middle of the stage and inside that house was Tim Berners Lee, the creator of the web, sitting at a desk in front of a computer. The script called for him to type a message in a web page which was then lit up in the seats around the arena. That message read:

"This (meaning the web) is for everyone."

When Berners Lee first announced the web, he also announced that it was being released without any use or licensing restrictions.

Aside from the corniness of the setup at the Olympics it's hard to overstate how big a deal this was. It's taken me a long time to really appreciate what those actions represented. He put the means to deploy an asynchronous and global network of linked

documents in the hands of everyone. For free.

When you consider the media landscape that both predates the web and that continues to dominate our lives, one defined by a handful of companies exercising a disproportionate influence on what can be said, when it can be said and by whom it is hard not to understand the web as an intensely political act.

Negative: San Francisco International Airport (SFO), United Air Lines, Douglas DC-8-50 *1964*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2011.032.1256***

It is worth remembering that the internet, and the web in particular, "happened" to the media and technology companies that existed at the time.

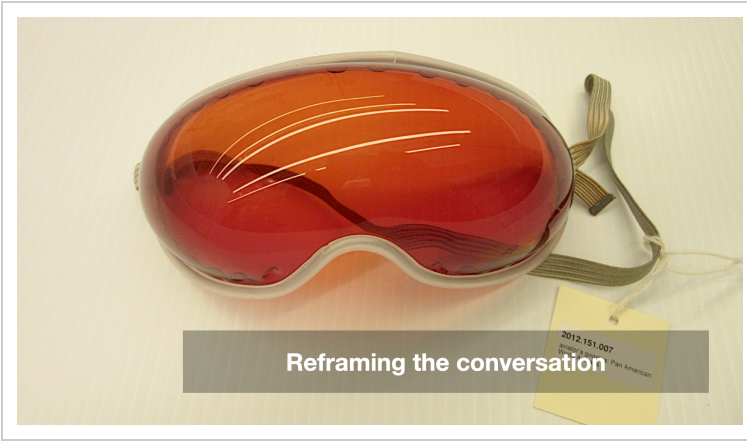
In as much as any of them were imagining what a universe of networked computers might look like they did not imagine the web. What they imagined, what they were actively promoting and marketing, were a series of branded communities, of walled gardens, with only limited means to interact with one another.

When the web first landed in people's lives neither Apple nor Microsoft had native TCP/IP stacks, TCP/IP being the networking layer that makes the internet work. People had to install third-party TCP/IP software (using floppy disks or some other kind of removable media) just to make the web work but they did, in droves, and it wasn't long before that kind of functionality became table stakes for all operating systems.

Since then the old media companies and the newer technology companies, which have become increasingly indistinguishable from one another, have embraced the web after a fashion but only as a kind of stepping-stone to bring people back to in to the fold of proprietary wall gardens.

Case in point: Mobile phones and the "app economy", an environment controlled by exactly two companies and designed to extract a commission from almost every interaction and to promote native and not-portable applications over web applications. But we also see the same behaviour from so-called "born of the web" companies like Facebook who have explicitly monetized reach, access and discovery. Facebook is also the company that gave the world React which is difficult not to understand as deliberate attempt to embrace and extend, to redefine, HTML itself.

Perversely, nearly everything about the mobile/app economy is built on, and designed to use, HTTP precisely because it's a common and easy to implement standard free and unencumbered by licensing.



Goggles: Pan American World Airways *1950s*
Gift of William E. Talbott
Collection of SFO Museum ***2012.151.007***

The recent history of the web, of the internet, has largely been defined by companies engaged in various forms of extractive behaviour, first with their customers but increasingly amongst themselves. Facebook for example is still trying to recover from, and adapt their business model to, Apple's decision to deny them access to the vast swathes of customer data that previously they harvested from people's iPhones and resold.

Which brings us to virtual, extended or augmented reality and the so-called "metaverse". There are lots of different, valid ways to think about this idea of extending space, activity and community in to virtualized, generative and networked environments. The possibilities are, both literally and figuratively, endless.

But it would be naive not to understand the recent push towards virtual technologies, advanced by the same companies that have dominated the internet landscape for the last decade as anything

but more of the same, just with a hypothetical reshuffling of the gate-keepers.

If it is already hard to complete with companies like Facebook or Instagram, because they enjoy such commanding slices of the marketplace, what "VR" heralds is more of the same but with a technical footprint so expensive and so demanding that it all but ensures it will only be within the means of a very few companies to operate.

Apple's recently announced Vision Pro headset is, by all accounts, a marvel of hardware design and software integration. It is Apple at its very best. It also represents 20, probably even 25 years, of ongoing research and development and product launches. Look at the Vision Pro device and then look at the last ten years of hardware and software releases and it is nearly impossible not to see the outlines of the device taking shape in plain sight.

It is also one of the most terrifying pieces of technology I've seen in the last decade. When you stop to consider everything it claims to do, it is a near-perfect surveillance device and the history of miniaturization in electronics coupled with the push towards wearable and embedded computing suggests, at the very least, a haunting future on the horizon.

Honestly, the only thing to save this device is the trust that Apple has marketed its brand on over the last few years. But a brand promise is about as fleeting a guarantee as you can get and there is a limit to the kinds of hardware controls you can bake in to these devices to ensure that promises of security and privacy are kept. Any sensor, or recording device, connected to the internet will forever be a tempting fruit.

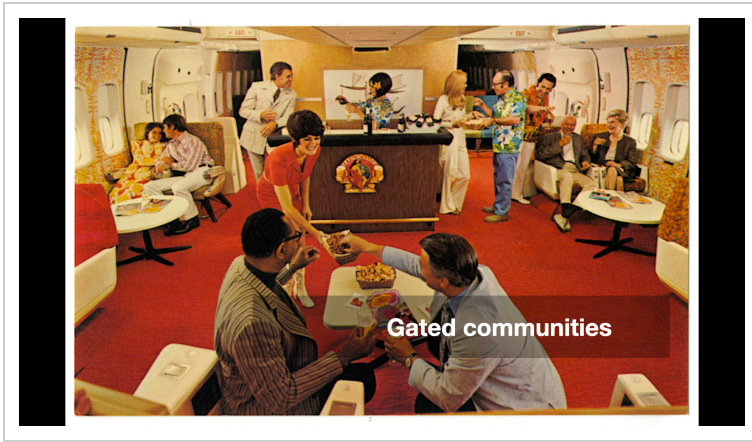
Speaking of which since the launch of the VisionPro, Facebook has updated their own virtual reality headset to include front-facing cameras to enable the same kind of "pass-through" functionality that the Vision Pro offers. "Pass-through", as the name implies, allows you to see the physical world around you in concert with augmented, or mixed, reality experiences.

But who here believes that Facebook doesn't have other motives for enabling an always-on camera capturing the world around you? In case you need any reminding this is the company that gave the world the Cambridge Analytica scandal which, really, is just the tip of the iceberg.

Facebook it is also worth noting has, in the span of just three years, hired and then fired 10,000 people to work on their "metaverse" initiative. It has reportedly spent upwards of a billion dollars already and, still... they haven't figured out how to render legs faithfully.

Meanwhile, Microsoft has gotten out of the consumer virtual reality headset market entirely. Snap recently shut down their commercial services AR division. Even Google has pulled back from headsets in the years since the launch of the ill-fated Google Glass.

It remains a challenging space. If any company is able to realize the technical vision of virtual reality and make it compelling enough for people to use then the rewards stand to be significant but, as I mentioned, not many companies will succeed and the world we will live in will look very much like the world of the big broadcast networks that existed before the web.



Postcard: Continental Airlines, Boeing 747 *1970s*
Gift of Thomas G. Dragges
Collection of SFO Museum ***2015.166.0551***

So let me be blunt:

The cultural heritage sector, on the other hand, can barely afford to hire and retain the staff necessary to keep a single website running for more than five years, let alone a decade or the twenty years we'll imagine it took to launch the Vision Pro headset.

It stands to reason then that in as much as we choose to engage with these new platforms it will never be on our terms and it will only ever be a commercial and a rent-seeking transaction.

Teapot: airplane motif *1935*
Purchase
Collection of SFO Museum ***1995.22.01 a b***

Historically, the museum sector has used technology as a proxy to validate its efforts. We keep an eye on what's "cool" and "new" and then we integrate those technologies in to our programming as a way to stay, or at least appear, relevant.

In 2023, the web is not "cool" anymore. It's not clear whether extended reality technologies are cool yet but there is definitely a concerted effort to try and make them seem that way. The point I have been driving at in this talk is that a funny thing happened thirty years ago:

The web happened and while it was absolutely the new, shiny, cool thing at the time it also happened to be the technology that most closely aligns, by design and by intent, with the purposes and motivations of the cultural heritage sector.

The web is the means by which the acts of revisiting and recall of

our collections, our programming and our institutional histories have become technically feasible, economically viable and with a reach and on a schedule that has literally never before been possible.

We would do well to recognize that. We would do well to understand the web not just as a notch in the linear progression of technological advancement but, in historical terms, as an unexpected gift with the ability to change the order of things; a gift that merits being protected, preserved and promoted both internally and externally.

The point is not that our relationship with technology should end with the web. The point is the web allows us to reframe our relationship with technology. Importantly, it enables – it does not guarantee, but it enables – us to reframe our relationship and dependence on the providers of those technologies.

The challenge of that relationship lies in the fact that as often as not the motivations of those providers and their platforms are not our own. Yet we continue to make an increasingly Faustian bargain to engage with them because we believe that these places are where our audiences have gone. Sooner or later that debt will come due so we would do well to recognize that the alternative, and a good alternative at that, is within our grasp.

Negative: San Francisco Airport, Terminal Building construction *1952*
Transfer from San Francisco International Airport
Collection of SFO Museum ***2011.032.0303***

Failing that, maybe the passing of the web can at least be the prompt we need to stop pretending that our relationship with technology is anything but window dressing.

Thank you.

2023-11-11

things I have written about elsewhere #20231129

Experiments in Photogrammetry

Experiments in Photogrammetry

Negative: San Francisco International Airport (SFO), film crew. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.2109***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/11/29/3d/` , in November 2023.

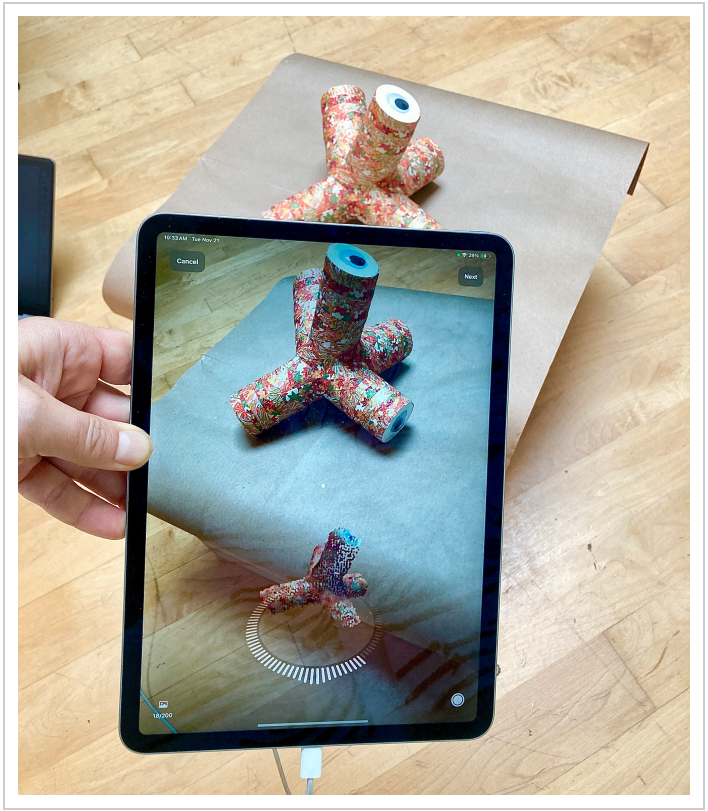
This is a blog post about using consumer-grade hardware and a suite of open-source software tools to generate high-quality three-dimensional (3D) models from photographic imagery, a process commonly referred to as "photogrammetry", which can be viewed and interacted within a web browser. This is a fairly technical blog post but the non-technical takeaways are: It's possible, it's affordable or at least meaningfully cheaper than it used to be, it's harder than it should be, we have developed tooling to make things easier and we would love your help to make it all better.

The catalyst for this work was the **Meet Object Capture for iOS** `https://developer.apple.com/wwdc23/10191/` session at Apple's 2023 developer conference. This session introduced people to the ability for iOS devices equipped with LiDAR cameras to not only capture

the imagery used to derive 3D models but also to render those models "on device". In 2023 an "iOS device equipped with LiDAR camera" can cost as little as $600 for a refurbished iPhone 12 Pro or $800 for an entry-level iPad Pro. As part of the developer session Apple provided a **sample application** `https://developer.apple.com/documentation/RealityKit/guided-capture-sample` demonstrating how to use these new features. We have taken that application and used it as the first building block in what we imagine could be an efficient pipeline for capturing and publishing 3D models within the operation and financial constraints of a museum. This blog post outlines that work describing the hiccups we encountered along the way and how we addressed those problems.

Before going any further I want to mention that the object used to test photogrammetry tools in this blog post is *not* an object from the **SFO Museum Aviation Collection** `https://collection.sfomuseum.org` . It's an object I brought in from home and it was chosen because it is a complex shape with lots of textures and odd surfaces. In short it's an object that is more like a lot of the things in a museum collection than not and highlights some of the challenges (or at least considerations) that remain when creating 3D objects. In fact, the final model used for this blog post is incomplete precisely because I wasn't very careful when considering which angles and surfaces to scan. I chose to leave the incomplete model as-is to demonstrate the results of that inattention.

Capturing



As mentioned, capturing imagery to create 3D models is done using a fork of Apple's **Scanning objects using Object Capture** `https://developer.apple.com/documentation/RealityKit/guided-capture-sample` example application included with WWDC23 session 10191: **Meet Object Capture for iOS** `https://developer.apple.com/wwdc23/10191/` . This is a very good application providing an intuitive interface, useful feedback and helpful prompts about lighting and indicating which parts of an object need to be scanned in order to produce a good model.

The motivation behind our fork is to use the existing application for

scanning objects but to update and add functionality for transferring (scan) data to a MacOS device for rendering higher-quality models, described below. As of this writing our application contains the following changes:

1. ***It adds an `Info.plist` property to enable iTunes file sharing so that scan data can be transferred from an iOS device to a MacOS device.***

2. ***It explicitly disables removing scan data*** `https://github.com/sfomuseum/ios-guided-capture/blob/main/GuidedCapture/AppDataModel.swift#L327-L333` ***(images and other associated files) when an object scan is completed or cancelled. That means it is left to individual users to remove scan data from iOS devices as necessary.***

3. ***It adds stub (disabled) code to trigger confirmation dialogues to signal whether on-device rendering should occur and/or whether scan files should be removed. Neither of these things work yet. This is a SwiftUI application and I am not a SwiftUI person so I am still feeling my way around things. As of this writing I have exhausted the time available for an exploratory project. The changes described above are not sufficient for putting the application in the hands of other museum staff members but they are "good enough" to move things forward.***

We have not published a "finished" version of this application so if you want to try it out you will need to download **the code** `https://github.com/sfomuseum/ios-guided-capture` and compile it locally.

Rendering

After running the "guided capture" application I have a finished model that I can view and interact with on-device. I can even export the finished model but because it was rendered on an iOS device it is created with a lower resolution ("preview") than the museum would want for this kind of work.

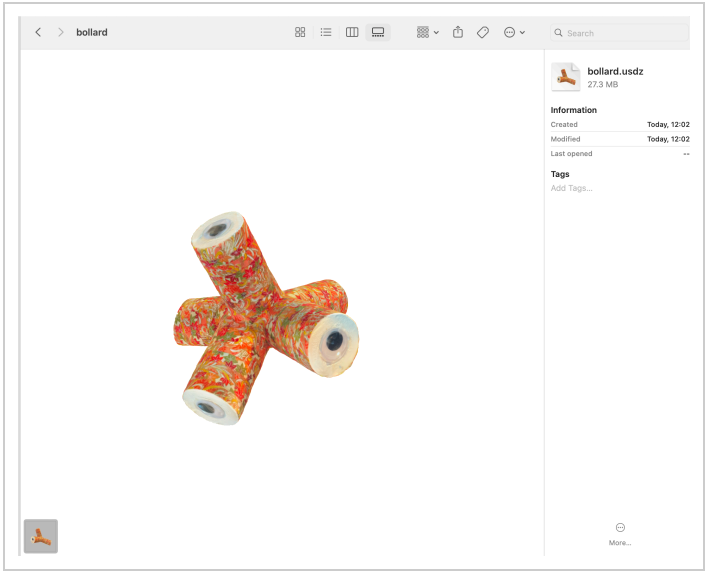
We have developed a Swift **library** `https://github.com/sfomuseum/swift-photogrammetry-render` and **command-line tool** `https://github.com/sfomuseum/swift-photogrammetry-render-cli` to render a set of scanned images at any of the available resolutions. There is also a good MacOS application called **Photogrammetry** `https://github.com/Unbinilium/Photogrammetry` for doing the same thing with a graphical user interface but we developed our own library and command line tool because we imagine integrating it in to an automated (and "headless") pipeline for processing 3D models.

The **command line tool** `https://github.com/sfomuseum/swift-photogrammetry-render-cli` takes as its input the path to a folder containing images of a scanned object, the path to the 3D model derived from those images and a set of optional flags to specify how the model should be created. In this example I am creating a model with a resolution of "medium" which will produce a 27MB file; by comparison a model rendered with "preview" level resolution will be about 2.5MB and a model rendered with "full" resolution will be over 70MB.

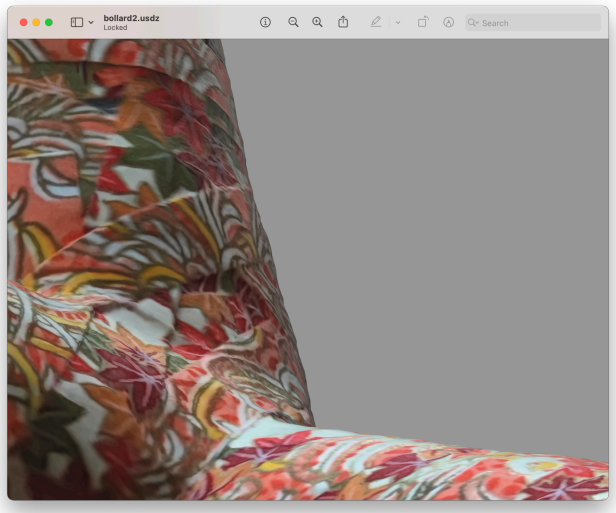
```
$> cd swift-photogrammetry-render-cli
$> swift build

$> ./.build/debug/render \
        /usr/local/data/Scans/2023-11-20T19:18:18Z/Images/ \
        /usr/local/data/bollard.usdz \
        --detail medium
```

Remember: After completing the image capture and rendering the object on iPad (shown above) I have connected that iPad to a MacOS device and transferred the images manually in to the `/usr/local/data/` folder.

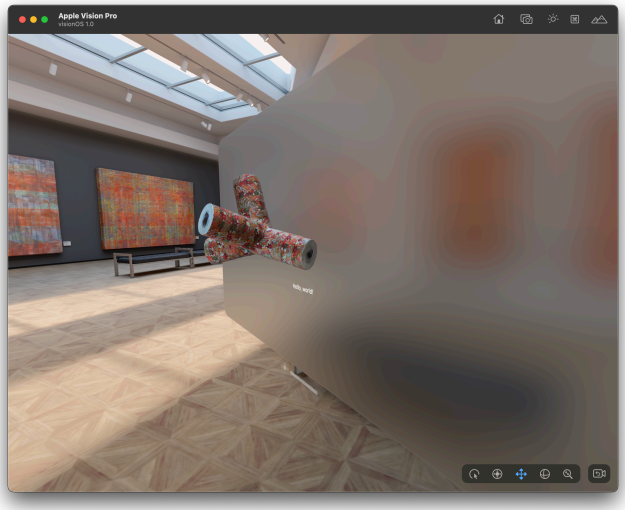
After the model is created I can view and interact with in the MacOS `Finder` application.



Here's that model zoomed in to a detail in the `Preview.app`. This model was derived from images captured using the camera on an iPad. The fact that a (lower-resolution) model can also be generated on that same iOS device suggests that **the upcoming Vision Pro headset** `https://www.aaronland.info/weblog/2023/11/11/therapy/#wishful` will be capable of continuously "photogrammetry-ing" everything around it which is both fascinating and a little terrifying.



These models are gaining support throughout all of Apple's operating systems and applications. For example, I can load these models in to `Keynote` presentations or even in to `VisionOS` applications developed for the Vision Pro headset. Here's a screenshot of the latter running in `XCode`'s Vision Pro emulator which conveniently ships with a "museum" environment:



On the other hand, I can not **view these models in a web browser** `https://github.com/immersive-web/model-element` "as-is" yet.

USDZ

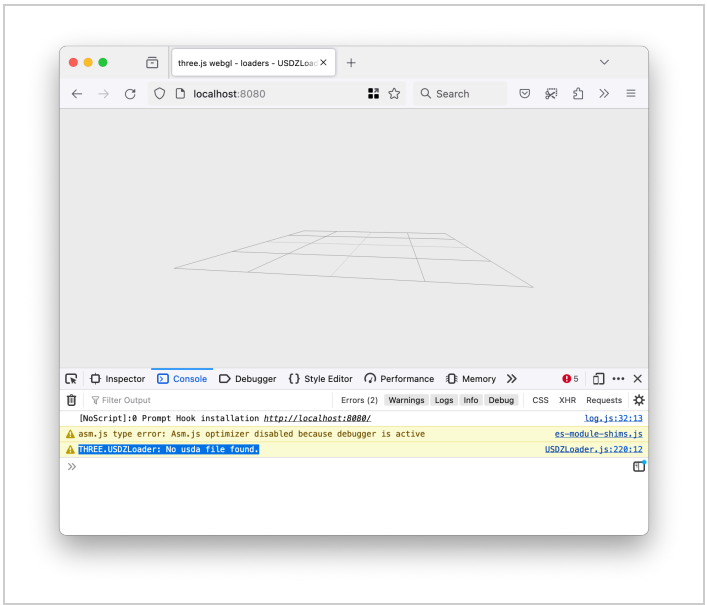
There are a lot of different file formats for 3D models but this is not the time or the place to discuss their relative merits. Apple, for its own reasons, has chosen the **USDZ** `https://developer.apple.com/documentation/realitykit/realitykit-usd-assets` format which is a variant of Pixar's **Universal Scene Description** `https://github.com/PixarAnimationStudios/OpenUSD` (USD) format.

Likewise, there are a number of different frameworks for rendering 3D models in a web browser but for the purposes of this blog post we started

with a framework called **three.js** `https://threejs.org/` which has been around for a number of years and supports a variety of 3D model formats including USDZ models. For example, here is some abbreviated JavaScript code to do just that:

```
const loader = new USDZLoader();

loader.load('./bollard.usdz', function (usd){
        scene.add(usd);
});
```

But when I try to run that code in a web browser this is what I see:



Specifically, this error message:

```
THREE.USDZLoader: No usda file found.
```

It turns out that USDZ files can contain either a "USDA" or a "USDC" mesh. For example, here are the contents of a "USDZ" model produced by

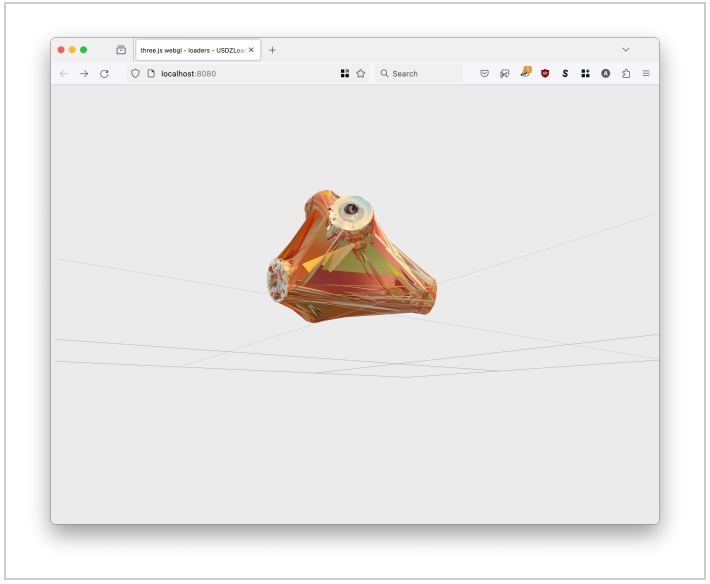
Apple:

```
M    Mode       Size         Date&time          Filename
- ----------  --------  -------------------  ----------------
  -rw-rw-rw-   2690955  21-Nov-2023 11:22:22  baked_mesh.usdc
  -rw-rw-rw-  26301765  21-Nov-2023 11:22:18  0/baked_mesh_disp0.exr
  -rw-rw-rw-  22684331  21-Nov-2023 11:22:18  0/baked_mesh_tex0.png
  -rw-rw-rw-   9811748  21-Nov-2023 11:22:18  0/baked_mesh_norm0.png
  -rw-rw-rw-   4326353  21-Nov-2023 11:22:18  0/baked_mesh_ao0.png
  -rw-rw-rw-   9092442  21-Nov-2023 11:22:18  0/baked_mesh_roughness0.png
- ----------  --------  -------------------  ----------------
              74907594                        6 files
```

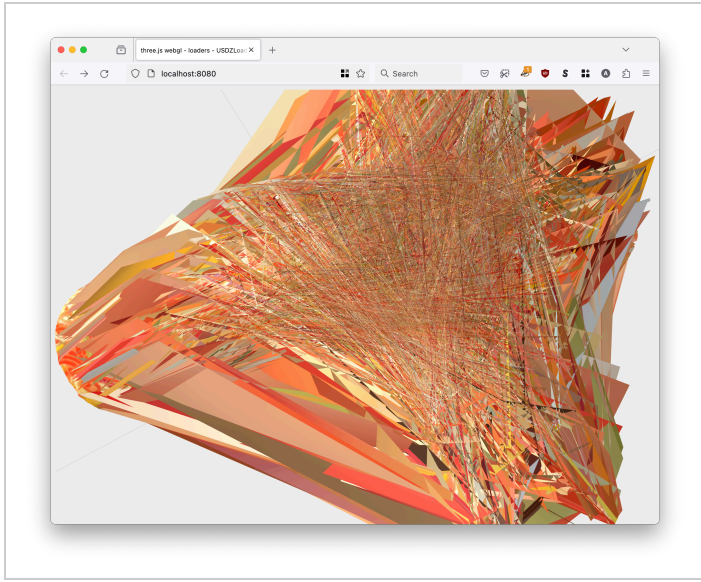
At this point, I'll confess that I don't really understand what the difference is between a "USDA" and a "USDC" mesh except that Apple uses the latter and everyone else, it seems, expects the latter. During my investigations, I did discover that Pixar has released **command line tools for converting between the two** `https://openusd.org/release/toolset.html#usdcat` . These tools require time and patience to install so what follows is not a recommendation to use them (for reasons that will become clear shortly) so much as a demonstration that it is, technically, possible to convert a USDZ model generated by Apple in to a USDZ model that can be consumed by everyone else.

```
$> mkdir bollard
$> cp bollard.usdz bollard

$> cd bollard
$> unzip bollard.usdz

$> /opt/local/USD/bin/usdcat baked_mesh.usdc -o baked_mesh.usda

$> zip -r ../bollard-a.usdz ./
```

Which all works fine until I view the new model which ends up being rendered like this:



While this model has a certain quality and could be an interesting jumping-off point for some **larger discussions about representation, interpretation and mediation** `https://new-aesthetic.tumblr.com/` it's really not going to be suitable for the purposes of a museum collection:



glTF

Did I mention that there are a lot of 3D object model formats? In addition to USDZ there is another one called **glTF** `https://en.wikipedia.org/wiki/GlTF` which is becoming a common standard for, it seems, everyone except Apple and is well-supported in web browsers. But how do we convert our USDZ model in to a glTF model? By benefitting from the hard work of all the people who've contributed to the **Blender** `https://www.blender.org/` project and Rob Crosby who wrote a Blender plugin to **import and export the wide range of USDZ models** `https://github.com/robmcrosby/BlenderUSDZ` .

Blender is an open source 3D modeling application `https://www.blender.org/` that supports plugins and scripting which can be run in "headless" mode from the command line. In short, we can combine all of these things alongside **a custom script to import a USDZ model and then export it as a glTF**

model `https://github.com/sfomuseum/BlenderUSDZ/blob/main/scripts/usdz2glb.py` like this:

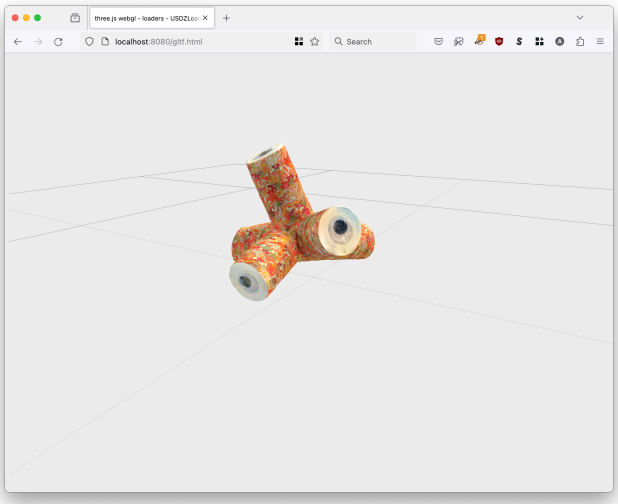
```
$> blender --python \
        /usr/local/sfomuseum/BlenderUSDZ/scripts/usdz2glb.py \
        /usr/local/data/bollard.usdz \
        /usr/local/data/bollard.glb
```

A quick reminder: The reason we want to be able to script these things and run them from the command line is that our larger goal is to automate this entire process so that the output of the `render` *tool, described above, can be fed to our* `usdz2glb.py` *Blender script to produce a glTF model without any manual intervention.*

And then if we update the `three.js` code to import and render a glTF model (instead of a USDZ model) like this:

```
const loader = new GLTFLoader();

loader.load('./bollard.glb', function (gltf){
        scene.add(gltf.scene);
});
```

Our model now appears in a web browser:



Using Blender to do this work feels a little bit like lighting a match with a tank but it does seem to "just work" and it (Blender) is supported in both Linux and Windows environments which means this piece of the pipeline we're imagining could happen **in the "cloud"** `https://github.com/nytimes/rd-blender-docker` or some other set of computing resources not limited to Apple hardware.

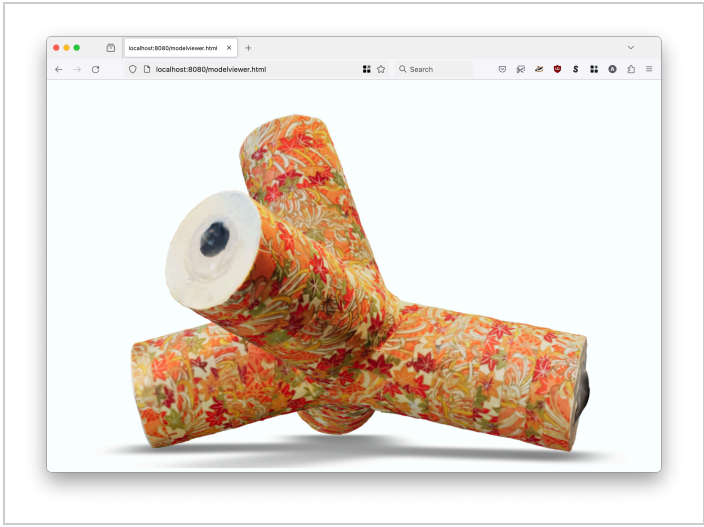
Model Viewer

Recently, Google has introduced the **`model-viewer` Web Component** `https://modelviewer.dev/` for declaring and rendering 3D models (or at least glTF models) in just a couple of lines of HTML and JavaScript code. For example this code:

```
<html>
    <head>
        <script type="module" src="model-viewer.min.js"></script>
        <style type="text/css">
         model-viewer {
                height: 100%;
                width: 100%;
         }
```

```
        </style>
    </head>
    <body>
        <model-viewer src="bollard.glb" ar environment-image="" shadow-intensity="1" came
    </body>
</html>
```

Yields the following in a web browser:

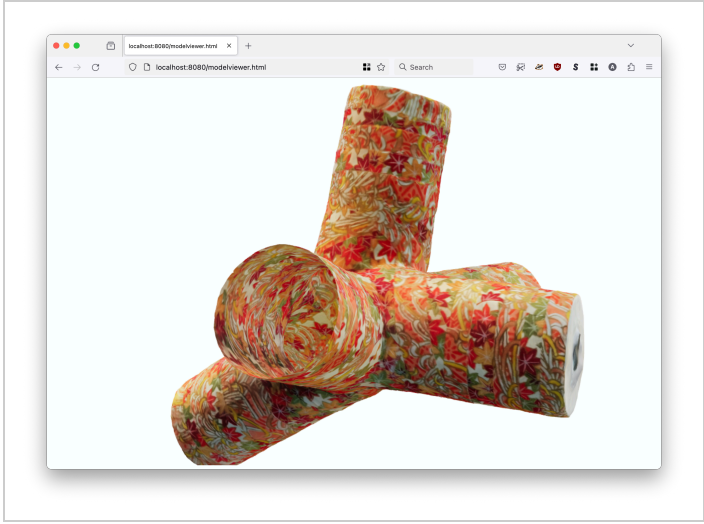


Which is pretty great! `three.js` is an incredibly powerful framework for working in three dimensions in a web browser but because it is can do so many things it is also a bit "too much" for simply rendering models in place and the `model-viewer` Web Component makes everything just a little bit simpler. So much so that we've included a `model-viewer` element for the model discussed throughout this blog post below:

Note: For reasons I don't completely understand I can not get this model to load on my own site. For the time being ***you can see it over here*** `https://millsfield.sfomuseum.org/blog/2023/11/29/3d/` *. Computers, amirite?*

Note: The W3C is developing the specification for ***a standard*** `model` ***HTML element to display 3D content*** `https://github.com/immersive-web/model-element` *across all browsers but it might still be a while before that work is completed.*

Next Steps (Help Wanted)



This blog post describes everything we've done so far. There are a number of things to be improved including, but not limited to:

ios-guided-capture

1. *Modal dialogs or a settings panel to indicate whether a model should be rendered on device.*

2. *Modal dialogs or a settings panel to indicate that source imagery should not be removed when a new scan is started.*

3. *How do you integrate **settings panels (bundle)** `https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/UserDefaults/Preferences/Preferences.html` in a SwiftUI application?*

4. *Controls to bundle (compress) scan imagery and transfer (upload) to a remote source (or at least via the native **share menu** `https://www.hackingwithswift.com/articles/118/uiactivityviewcontroller-by-example` controls for easier transferring via AirDrop, etc.)*

swift-photogrammetry-render (-cli)

1. *Improved documentation*

2. *Tests*

BlenderUSDZ scripts

1. *Improve the **custom "export as gltf" Blender script** `https://github.com/sfomuseum/BlenderUSDZ/blob/main/scripts/usdz2glb.py` that we wrote. Specifically this script represents the minimum amount of code to "make it work" without really understanding either Blender or the*

Blender Python API in any great detail. I find it hard to believe that I found the most robust way to do things on the first try.

The Code

We would welcome pointers, suggestions and contributions for any of these items. Our goal is, first, to have a suite of tools that SFO Museum can use to quickly, easily and affordably produce 3D models of its collection objects but also, as part of our commitment to **"small focused tools"** `https://millsfield.sfomuseum.org/blog/2020/08/04/small-tools/` to publish this work as discrete reusable (and rearrange-able) components for use in a variety of circumstances by a variety of cultural heritage organizations.

> The cultural heritage sector needs as many small, focused tools as it can produce. It needs them in the long-term to finally reach the goal of a common infrastructure that can be employed sector-wide. It needs them in the short-term to develop the skill and the practice required to make those tools successful. We need to learn how to scope the purpose of and our expectations of any single tool so that we can be generous of, and learn from, the inevitable missteps and false starts that will occur along the way.

All of the code described in this blog post is available from the **sfomuseum GitHub account** `https://github.com/sfomuseum` :

1. ***https://github.com/sfomuseum/ios-guided-capture*** `https://github.com/sfomuseum/ios-guided-capture`

2. ***https://github.com/sfomuseum/swift-photogrammetry-***

render `https://github.com/sfomuseum/swift-photogrammetry-render`

3. ***https://github.com/sfomuseum/swift-photogrammetry-render-cli*** `https://github.com/sfomuseum/swift-photogrammetry-render-cli`

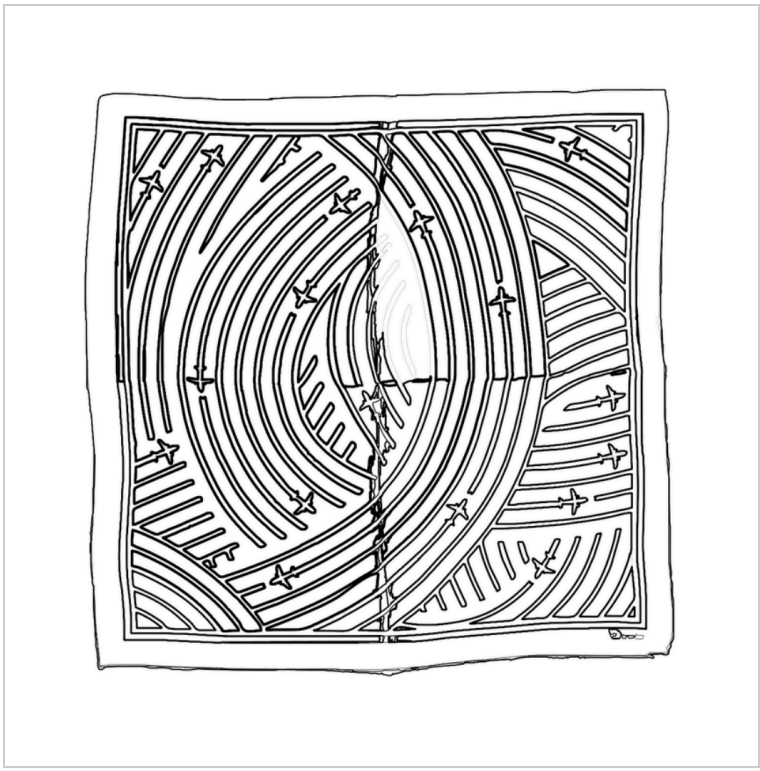
4. ***https://github.com/sfomuseum/BlenderUSDZ*** `https://github.com/sfomuseum/BlenderUSDZ`

2023-11-19

things I have written about elsewhere #20231218

Coloring Books Pages For the SFO Museum Aviation...

Coloring Books Pages For the SFO Museum Aviation Collection

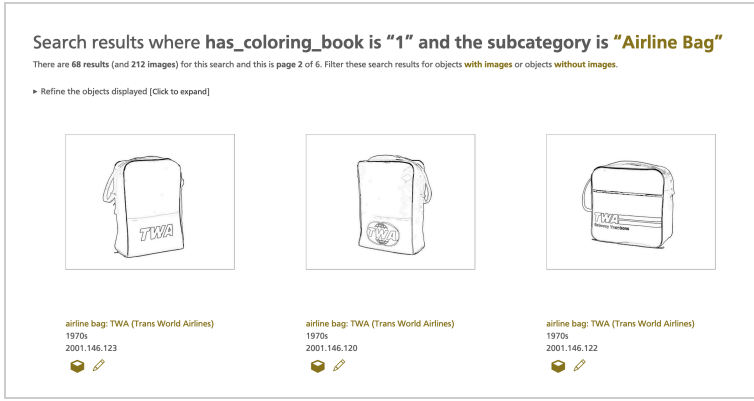


Flight attendant scarf: Virgin America. Silk. Gift of Sirena Lam, SFO Museum Collection. ***2019.032.090***

This was **originally published on the SFO Museum Mills Field weblog** `https://millsfield.sfomuseum.org/blog/2023/12/18/coloringbook/` , in December 2023.

We have launched a new experimental feature on the SFO Museum Aviation Website: Coloring book pages (or sheets) for a subset of the objects in our collection. As of this writing those items are **model airplanes** `https://collection.sfomuseum.org/search/advance`

d/?has_coloring_book=1&subcategory=1511213161 , **scarves** https://collection.sfomuseum.org/search/advanced/?has_coloring_book=1&subcategory=1511213449 , **airline bags** https://collection.sfomuseum.org/search/advanced/?has_coloring_book=1&subcategory=1511213155 and a small number of other objects that were added during testing.



You can see all of the objects with coloring book pages at:

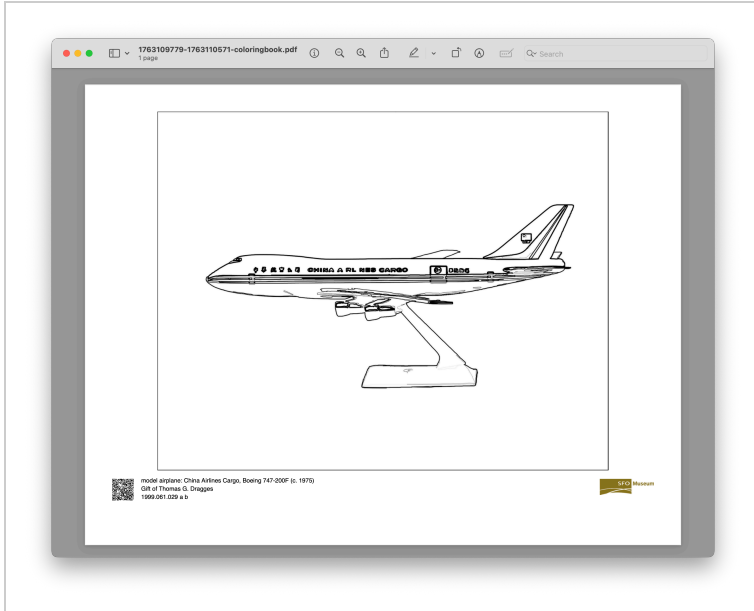
https://collection.sfomuseum.org/coloringbook

Objects with coloring book pages are displayed with a icon underneath their title. Clicking on the icon will open a PDF file in a new window with a stylized, black and white outline of that object for you to print out and color as you see fit. For example, the coloring book page for **this object** https://collection.sfomuseum.org/objects/1763109779 :

Model airplane: China Airlines Cargo, Boeing 747-200F. Plastic, paint. Gift of Thomas G. Dragges, SFO Museum Collection. ***1999.061.029 a b***

Looks **like**

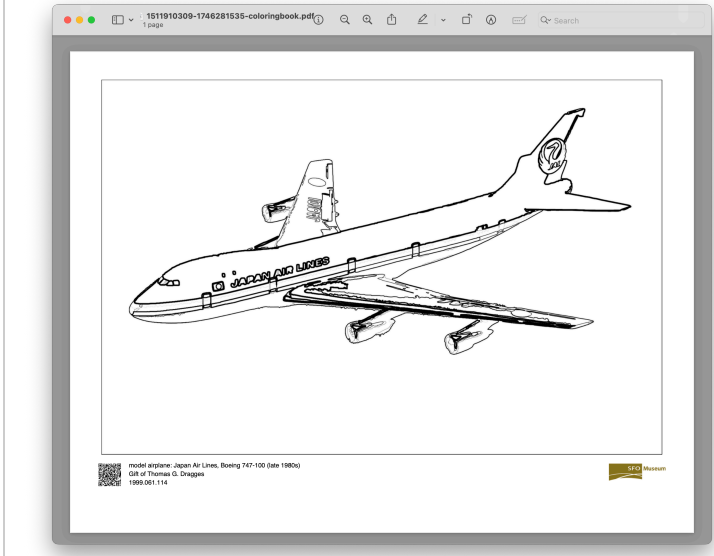
this `https://static.sfomuseum.org/media/coloring/176/310/977/9/1763109779-1763110571-coloringbook.pdf` :



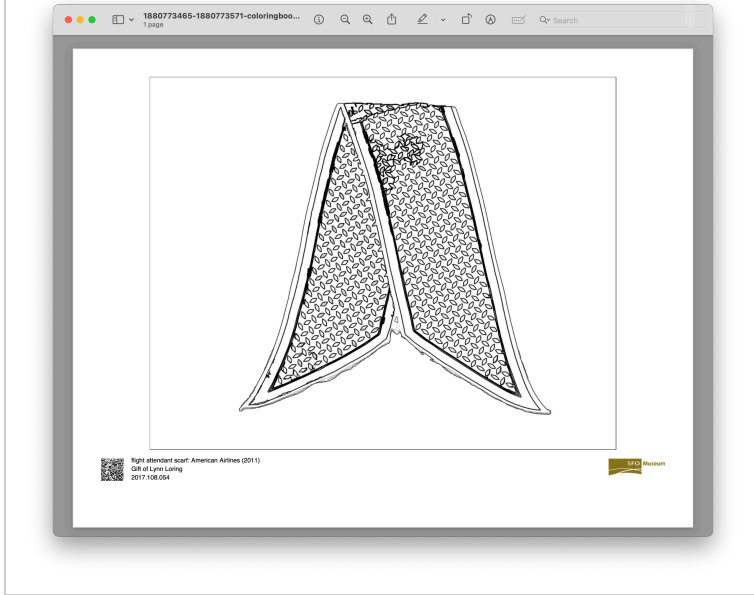
Each coloring page also has a QR code linking to that object on the **SFO Museum Aviation Collection website** `https://collection.sfomuseum.org` . As mentioned earlier this is still considered experimental work which means not all coloring book sheets are as good as others. Specifically: Sometimes the outlines are **too dense** `https://static.sfomuseum.org/media/coloring/176/327/874/1/1763278741-1763279877-coloringbook.pdf` and sometimes the outlines are **missing enough detail** `https://static.sfomuseum.org/media/coloring/188/026/367/5/1880263675-1880263705-coloringbook.pdf` . We've also seen some sheets where the object itself is drawn smaller than it should be. All of these things will be addressed in time but we are excited to tell you about this feature so that you and/or your children can get busy coloring!

In the **Updating the SFO Museum Wayfinding Service - Publications** `https://millsfield.sfomuseum.org/blog/2023/06/12/wayfinding-pdf/` blog post I wrote:

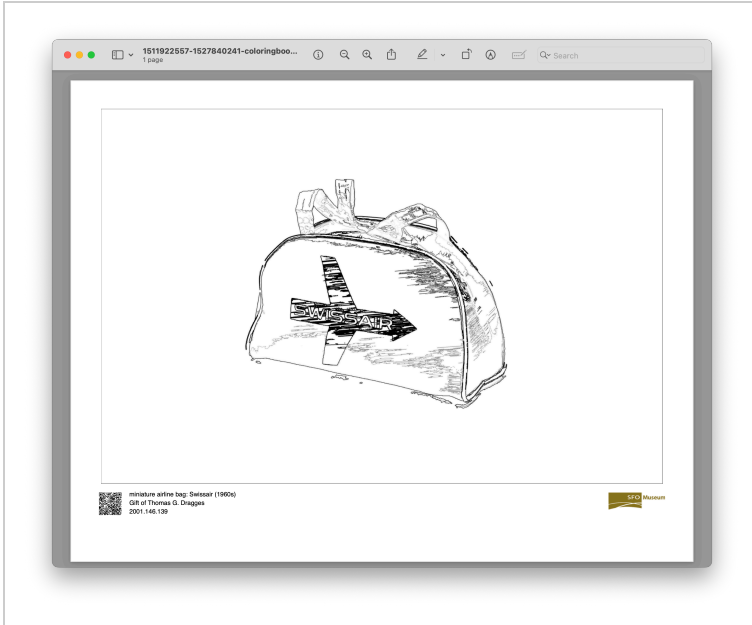
> Custom publications are a first attempt at designing and producing artifacts which can be thought to "follow a visitor out of the building" (or the museum (which also happens to be the airport)). Think of them as training wheels towards achieving that goal. We believe that the objects and public art works on display at SFO Museum and SFO are worth seeing in person but that doesn't, and shouldn't, preclude the ability to enjoy these things when you are not at SFO.



Model airplane: Japan Air Lines, Boeing 747-100. Plastic, paint. Gift of Thomas G. Dragges, SFO Museum Collection.
1999.061.114

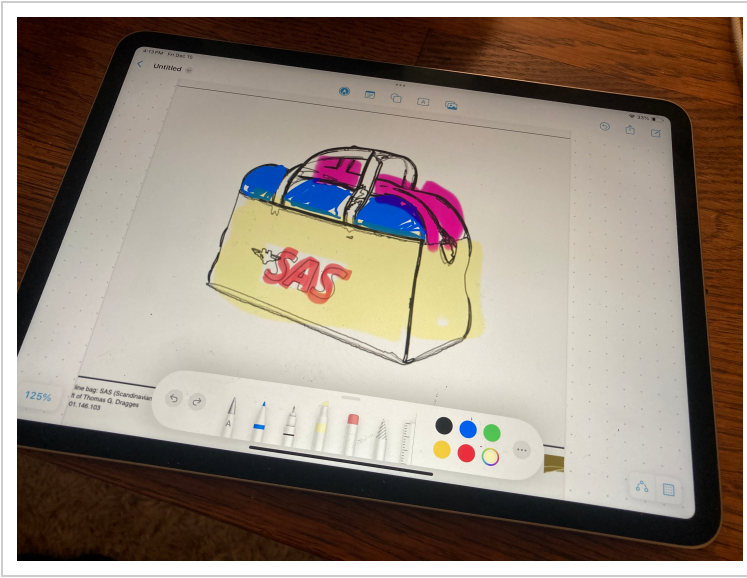


Flight attendant scarf: American Airlines. Fabric. Gift of Lynn Loring, SFO Museum Collection. ***2017.108.054***



Miniature airline bag: Swissair. Vinyl, plastic, metal. Gift of Thomas G. Dragges, SFO Museum Collection. ***2001.146.139***

Coloring book pages should be thought of as a second attempt at producing a similar artifact that can follow you "out of the building". In practical terms because there aren't public printer stations at the airport and because most people don't travel with a portable printer these artifacts are likely to be printed before a person even gets to "the building" (the airport) but you get the idea. These coloring book sheets can be downloaded to tablet devices and opened (and colored) in a variety of applications so if you are looking for something to do while you wait for your flight once you've arrived at the airport, or something to do while you're in the air, these can be a fun and creative activity.

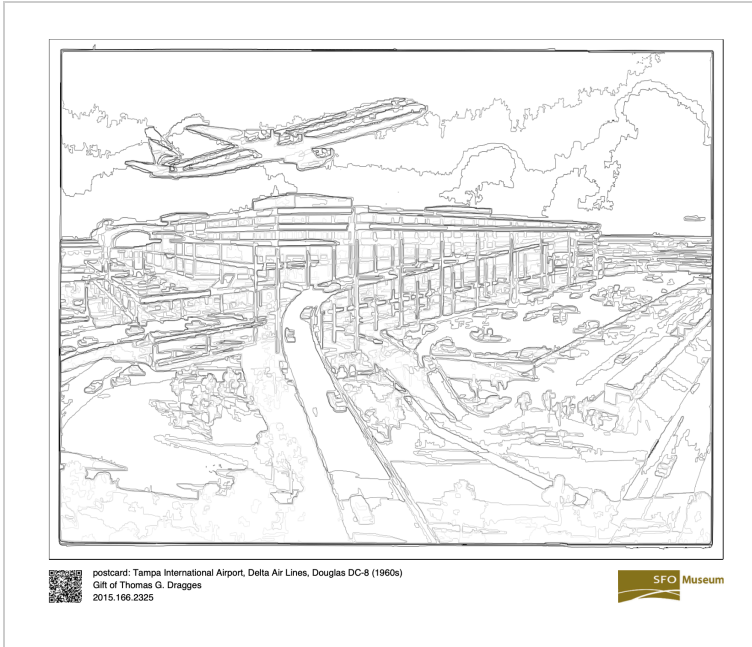


Airline bag: SAS (Scandinavian Airlines System). Synthetic textile, plastic, metal, cardboard, ink. Gift of Thomas G. Dragges, SFO Museum Collection. ***2001.146.103***

The rest of this blog post gets in to some of the technical details for how these coloring book sheets are produced. If that's not of interest or if you just want to start coloring you can stop reading now and start downloading coloring book sheets at:

https://collection.sfomuseum.org/coloringbook

The technical details



Postcard: Tampa International Airport, Delta Air Lines, Douglas DC-8. Paper, ink. Gift of Thomas G. Dragges, SFO Museum Collection. ***2015.166.2325***

Generating the outlines for coloring book sheets is a three-part process: First an image is simplified using the **visioncortex `vtracer` command line tool** `https://www.visioncortex.org/vtracer-docs` , which produces an SVG file. Second, that SVG file is turned back in to a raster image using the **`Batik` SVG rasterizer** `https://xmlgraphics.apache.org/batik/` . Finally, the outline is derived from that (second) image using **Michael Fogleman's `contourmap` application** `https://github.com/fogleman/contourmap` and saved as a third and final image which is embedded in a PDF document.

Sunglasses: Virgin America. Paper, ink. Gift of Sirena Lam, SFO Museum Collection. ***2019.032.103***

`vtracer` is written in the **Rust** `https://www.rust-lang.org/` programming language, `Batik` in **Java** `https://openjdk.org/` and `contourmap` in **Go** `https://go.dev` . Because we write most of our code in `Go`, including the code that produces and then publishes the final PDF files we include on the **collection website** `https://collection.sfomuseum.org` it would be convenient if the `vtracer` tool could output a raster image and if it could be called from library code in Go. At a glance **adding code to the vtracer library** `https://github.com/visioncortex/vtracer/blob/master/cmdapp/src/converter.rs#L29-L37` to rasterize its own SVG output shouldn't be too hard but I have only done **a little bit of Rust programming** `https://millsfield.sfomuseum.org/blog/tags/rustlang/` and I'll confess I got a bit lost trying to understand the documentation for the **resvg** `https://github.com/RazrFalcon/resvg` rasterizer. Likewise, while it appears possible to **call Rust libraries from Go** `https://words.filippo.io/rustgo/` it also looks sufficiently involved that it risks becoming a wild goose chase. Any help, or suggestions, with either of these things would be welcome.

The original version of this workflow used an SVG rasterizer written in Go but it was often very (very (very)) slow. `Batik` has been actively developed for nearly two decades, remains the gold standard for SVG rasterization and is fast so it wasn't a hard decision to adopt it until the `vtracer` tool can be updated to generate raster images.

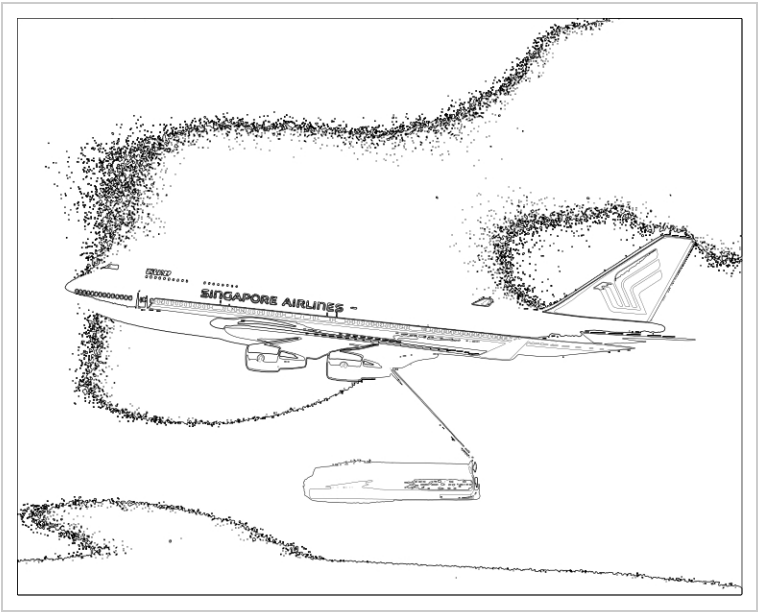
Negative: San Francisco International Airport (SFO), baggage carousel. Negative. Transfer from San Francisco International Airport, SFO Museum Collection. ***2011.032.0712***

That's a lot of moving pieces but it is possible to bundle them all in to a single Docker container which can be run from a **containerized AWS Lambda function** `https://docs.aws.amazon.com/lambda/latest/dg/images-create.html` so we can generate (and more importantly regenerate) coloring book sheets on demand quickly and inexpensively. Because of the complexities described we have not released any general-purpose tooling for this workflow but we have

published the code we use for people to consult as a reference, in particular **the steps to derive an outline for any given image** `https://github.com/sfomuseum/go-sfomuseum-coloringbook/blob/main/outline.go` :

https://github.com/sfomuseum/go-sfomuseum-coloringbook

Early on in the process we experimented with using **the `image-emboss` tool** `https://github.com/sfomuseum/swift-image-emboss-cli` that I wrote about in the **Extracting Subjects from Images in Swift (and gRPC)** `https://millsfield.sfomuseum.org/blog/2023/10/31/image-emboss/` blog post to do image segmentation to only derive outlines for the principal "subject" in an image. This was to account for the fact that while most of our collection images have backgrounds that are "just grey" they still have enough variation in them that, when they are used to derive contours, they produce a lot of distracting artifacts.



Model airplane: Singapore Airlines, Boeing 747 Megatop. Wood, metal, paint. From the estate of Louis A. Turpen, SFO Museum Collection. ***2023.068.044 a b***

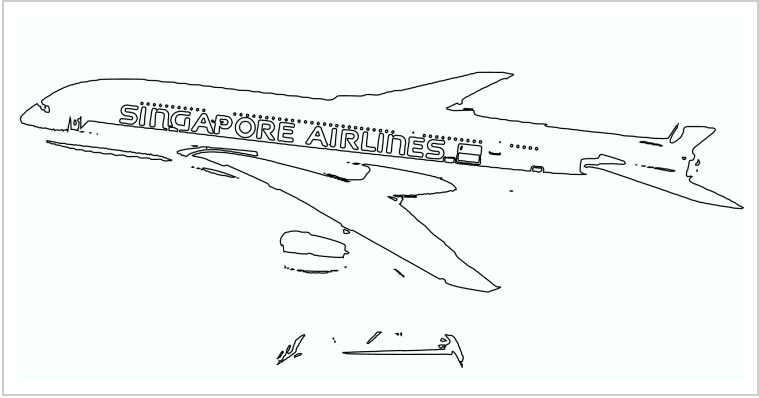
In **the blog post about the `image-emboss` tool** `https://millsfield.sfomuseum.org/blog/2023/10/31/image-emboss/` I wrote:

> Note that the underlying “subject lifting” code merges two objects in to one and excludes four others entirely. This is a good reminder that as impressive as the technologies used to perform these tasks are they remain fallible and the reasons they choose to do, or not do, something often remain opaque. While they herald exciting new possibilities they are still tools which should only be used with supervision and a good understanding (or at least awareness) of when and where they may not meet the needs of your specific project.

One of these "opaque decisions" manifested itself with this image of a **Braniff Airlines 727 plane** `https://collection.sfomuseum.org/objects/1511909345/` which when "embossed" (or "segmented") is missing an entire wing.

Model airplane: Braniff International, Boeing 727. Plastic, adhesive, paint. Gift of Ronald V. Wilson, SFO Museum Collection. ***1999.642.003***

Given that we always planned to include **model airplanes** `#` in our initial set of coloring book sheets these kinds of errors argued against using image segmentation without more investigation. We also considered using Apple VisionKit's **VNDetectContoursRequest** `https://developer.apple.com/documentation/vision/vndetectcontoursrequest` APIs but quickly determined that they have many of, if not the same, problems that `image-emboss` tool does. For example:



Model airplane: Singapore Airlines, Airbus A380. Plastic, paint. Transfer, SFO Museum Collection. ***2011.212.005 a b***

Eventually we settled on using the `vtracer` tool to flatten each image and accepting that a certain amount of background artifacting would be acceptable and might even present some interesting creative possibilities when outlined.

During this process I spent a little bit of time to determine whether we could use Stable Diffusion's **image to image tool** `https://huggingface.co/docs/diffusers/api/pipelines/stable_diffusion/img2img` to derive new images to use for creating outlines. For example, would it be possible to create images of specific airplane models or uniforms without any branding or **livery** `https://en.wikipedia.org/wiki/Aircraft_livery` ? It is worth noting that I didn't try very hard or train the model on more than a single image but the initial results were not promising. Given this image of **a Fiji Airways A320 aircraft** `https://collection.sfomuseum.org/objects/1763150591/` :

Model airplane: Fiji Airways, Airbus A330-200. Plastic, paint, paper, ink. Transfer from San Francisco International Airport, SFO Museum Collection. ***2016.026.056 a c***

And the prompt `“Model airplane of an unpainted Airbus A330-200 seen from nose to tail on a grey background”` the image that was produced was incorrect and incomplete, depicted an aircraft that is almost certainly not airworthy and either a grim prediction or a poignant commentary on **modern passenger seating** `https://en.wikipedia.org/wiki/Vertical_seat` .

So, there is work to do all around. There is work to do to better configure the settings we use to generate outlines using our current workflow in an automated fashion. There is work to do to better understand how we might use machine learning processes which probably means training our own machine learning models to use in that work. As always we welcome any suggestions and improvements and, in the meantime, happy coloring!

https://collection.sfomuseum.org/coloringbook

2023-12-18